VISIT…

Second Edition

Introduction to Network Analysis

Laura A. Chappell, Sr. Protocol Analyst
Protocol Analysis Institute, LLC
www.packet-level.com

- **Troubleshoot typical network problems on TCP/IP, IPX/SPX, Ethernet and Token Ring networks.**
- **Optimize network resource usage and design.**
- **Plan for future growth and identify true 'killer apps.'**
- **Save time and money by getting the packet-level answers to network problems.**
- **Target the most effective analysis solution for hubbed, switched, and routed networks.**
- **Get hands-on experience using "Laura's Lab Kit."**

The **ultimate reference guide** to using a protocol analyzer to troubleshoot and test network communications and applications.

This book provides prerequisite information for "Advanced Network Analysis" from *podbooks.com*.

Introduction to Network Analysis

(Second Edition)

Troubleshoot, optimize and test your network today!

Laura A. Chappell
Sr. Protocol Analyst
Protocol Analysis Institute, LLC
lchappell@packet-level.com

Published by

Introduction to Network Analysis
Second Edition
By Laura A. Chappell

Published by
podbooks.com
5339 Prospect Road, Suite 343
San Jose, CA 95129
USA

Copyright © 2001, podbooks.com. All rights reserved. No part of this book, including interior design, cover design and trace files, may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording, or otherwise) without the prior written permission of the publisher. Podbooks, AnalysisMan, AnalysisGal, and the AnalysisKids are trademarked by podbooks.com.

Library of Congress Catalog Card No.: Pending
ISBN: 1-893939-36-7
Printed in the United States of America (Saratoga, California)
10 9 8 7 6 5 4 3 2 1
Distributed worldwide through podbooks.com.

For general information on podbooks.com, including information on corporate licenses, book update procedures, and future titles, contact podbooks.com at 408/378-7841 or info@podbooks.com.

For authorization to photocopy items for corporate, personal, or educational use, contact the copyright division at copyright@podbooks.com.

Trademarks: All brand names and product names used in this book are trade names, service marks, trademarks, or registered trademarks of their respective owners. Podbooks.com is not associated with any product or vendor mentioned in this book.

Limit of Liability/Disclaimer of Warranty. Authors and publishers have used their best efforts in preparing this book. Podbooks.com and authors make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. There are no warranties that extend beyond the descriptions contained in this paragraph. No warranty may be created or extended by sales representatives or written sales materials. The accuracy and completeness of the information provided herein and the opinions stated herein are not guaranteed or warranted to produce any particular result, and the advice and strategies contained herein may not be suitable for every individual, especially individuals named 'Fred'. Neither podbooks.com nor author shall be liable for any loss of profit or any other commercial damages, including without limitation special, incidental, consequential, or other damages.

Copy Protection. In all cases, reselling or duplication of these materials without authorization is expressly forbidden.

podbooks.com
info@podbooks.com
www.podbooks.com

Protocol Analysis Institute
info@packet-level.com
www.packet-level.com

Welcome to podbooks.com

This is the fifth podbook released -- there are many more in the pipeline. We've changed the printing format - we've changed the desktop publishing format.

You'll notice many changes from the first version of this title to the current one. Especially the lab-based exercises listed in Chapter 5.

Pods often reference the Protocol Analysis Institute website - www.packet-level.com -- where we keep numerous trace files and technical materials that supplement the podbooks.

Pods still come in three formats -- an individual electronic (PDF) version, the hardcopy bound version and the multibook CD version. I highly recommend the electronic and CD version if you have a color printer -- the screenshots look far superior when viewed in color.

In May 2000, we videotaped my favorite seminars and bundled them with comprehensive workbooks for the "Laura Chappell presents... " seminars. The videos are uncut, unedited, unpolished - and often irreverent -- the only way to learn!

We are also very interested in your feedback on this and all other pods. Send your thoughts and comments directly to me at lchappell@packet-level.com. If you would like to receive release information about future projects, please join the mailing list at www.packet-level.com.

Laura A. Chappell
Founder, podbooks.com

About the Author

Laura Chappell, Sr. Protocol Analyst, is a highly-energetic speaker and author who has trained thousands of LAN/WAN administrators, engineers, technicians and developers. She admittedly 'lives, eats and breathes' packets and packet-level communications. She analyzes and documents network performance by providing onsite and offsite network analysis, troubleshooting and optimization services. Her international client base ranges from governments to health care to financial/banking institutions.

Ms. Chappell is the author of "Advanced Network Analysis Techniques," “TCP/IP Analysis and Troubleshooting” (podbooks.com), "Novell's Guide to LAN/WAN Analysis: IPX/SPX," "Novell's Guide to Internet Access Solutions," and "Novell's Guide to Multiprotocol Internetworking" (Novell Press/IDG Books/Sybex Books). Ms. Chappell has also edited three Cisco Press titles, "Introduction to Cisco Router Configuration," "Advanced Cisco Router Configuration," and "Cisco Internetwork Troubleshooting" (Cisco Press/Macmillan Technical Publishing).

In July 2001, Ms. Chappell will complete her first college textbook -- “Guide to TCP/IP” which features packet-level communications. This textbook will be distributed by Course Technology and is co-authored with Ed Tittel.

Ms. Chappell can be reached via email at lchappell@packet-level.com.

For Ginny and Scotty, my buddies.

About This Pod

This pod is the revised Edition of “Introduction to Network Analysis,” released by podbooks.com in 1999.

It was time to revise the text to show the more advanced analyzers that rule the marketplace and introduce some of their newer features.

Analyzers Used in this Book

Over the years, I have used numerous analyzer products -- from the early DOS-based analyzers that required manual decoding of the up-to-date protocols to the most recent graphical analyzers that illustrate the network traffic in methods too numerous to mention.

I’ve used good analyzers.

I’ve used bad analyzers.

I’ve used really lousy analyzers.

I hope you select an analyzer that can do all of the tasks shown in this book. The analyzers that I selected to develop this book with are the WildPackets EtherPeek and Network Associates’ Sniffer Pro. Both analyzers offer a wide range of features and are relatively easy to operate.

I feel it is important to show both products in this book -- as you will find out, each analyzer has its advantages and disadvantages. I use both analyzers in my daily work.

There are some add-on products that should be in your toolkit as well. We don’t just get along with the analyzers these days... I’ll introduce these tools and provide some details on how to use these products to enhance your analysis skills in Chapter 5.

Who Should Read This Book

This book is designed to support anyone who installs, configures, troubleshoots, and optimizes today's networks.

If you are a CNE, MCSE, CCIE or any other multi-letter geek, you should know your packets --- you should know how to analyze typical communications on the network.

Chapter Information

Chapter 1, "What is 'Analysis, Anyway?" covers the various elements of a network analyzer and provides instructions on tapping into hubbed, switched, or routed networks. In this section, you'll find out how standalone analyzers differ from distributed analyzers as well as the differences between software and hardware-based solutions.

Chapter 2, "How Do You Start?" walks you through the entire process of tapping in, obtaining a network overview (the 'Laying on of Hands'), classifying traffic and trends, identifying hot spots and documenting your findings. You'll learn how to read the summary window and detailed packet decodes. I'll introduce my favorite analyzers -- EtherPeek and Sniffer Pro. Finally, this chapter gives you a list of seven analysis tasks you can do today on your network.

Chapter 3, "Packet Formats and Decodes" takes you into the packet structures and communication sequences often seen on a network. In this chapter, we'll look the headers that are built to get packets from the application through the protocol stack and onto the network. Specifically, we'll examine Ethernet and Token Ring headers. This chapter also provides field offset and details for IP, TCP and UDP headers. Finally, we'll follow a typical communication to see how the data flows.

Chapter 4, "Identifying Typical Problems" focuses on the most common errors that occur on Ethernet, Token Ring, IPX/SPX and TCP/IP networks. After a brief overview of the communications processes used by each network type, you are introduced to the types of problems that you should watch for. Throughout this chapter, you are provided with extra links to areas that contain more depth of information… for those of you who just can't get enough.

Chapter 5, “Analysis Tools and Exercises” gives you an idea of the other tools that I use to analyze communications. These tools include my beloved NetScanTools Pro, PacketScrubber to sanitize packets, ProConvert to convert trace files between various analyzer versions, EtherPeek for Windows and Sam Spade. Of course, a good quality blender is also require to make the perfect Margarita!

Each chapter ends with a test to determine your comprehension level. The answers can be found in Appendix A.

Appendix A, "Answers to the Chapter Tests," provides the answers to all the questions posed at the end of the Chapters.

Appendix B, "The Basics of Data Flows," is a 'must read' for anyone who wants to troubleshoot or optimize network communications. It provides the foundation of knowledge on how packets are hubbed, switched and routed through a network. It covers what these interconnecting devices can and cannot do to the packet contents. You should know the materials in Appendix B inside and out.

Appendix C, "Resources and References," lists supplemental materials that can help you with your network analysis focus. Many of the materials were written by me (there seems to be a shortage of packet-level documentation these days). There are additional references and resources online at www.packet-level.com.

Table of Contents

Welcome to podbooks.com iii
About the Author iv
About This Pod vi
Table of Contents ix

CHAPTER 1 *What is &#*(#&!? is Analysis?* *1*

Network Analysis 101 2
Troubleshooting with an Analyzer 5
Optimization *7*
Planning and Testing with an Analyzer 8
What the Heck is a Network Analyzer? 10
Standalone v. Distributed Analyzers 12
How to Use a Standalone Analyzer *12*
How to Use a Distributed Analyzer *13*
Software v. Hardware/Software Analyzers 15
Analyzer Placement 16
Hubbed Network *16*
Bridged Network *16*
Switched Network *18*
Hubbing Out 19
Analyzer Agents 20
Port Spanning/Mirroring 20
Routed Network 21
WAN Link *22*
Full-Duplex Connections *24*
Analyzer Elements 25
Promiscuous Mode Card and Driver *26*
Decodes *26*
Capture Filter (Pre-Filters) *27*
Display Filters (Post-Filters) *27*
Gauges/Graphs *27*
Alarms/Alarm Thresholds *28*
Trace Buffer *30*
Your Basic Online Resource List 31
Protocol Analysis Institute 31
Internet Assigned Numbers Authority 31
Institute of Electrical and Electronic Engineers 31
Internet Engineering Task Force 31
Chapter 1 Test 32

CHAPTER 2 *How Do You Start?* *34*

Overview of the Process **35**
Identify Areas of Interest **36**
Tap In (and Optionally Redirect Traffic) **37**
Perform the 'Laying on of Hands' **38**
Classify Traffic and Document Trends **42**
Identify Possible 'Hot Spots' **44**
Focus in on Specific Conversations and Packet Types **45**
All There is to Know About Pattern Analysis **47**
Document Your Findings **49**
Reading/Understanding the Summary Window **50**
- *Marker Box Column/Marking Capability* ***52***
- *Number Column* ***53***
- *Status Column* ***54***
- *Source Address Column* ***54***
- *Source Port* ***55***
- *Destination Address Column* ***55***
  - MAC Layer Broadcasts 55
  - IPX Broadcast/Multicast Addresses 56
  - Internet Broadcast/Multicast Addresses 56
- *Destination and Source Ports* ***56***
- *Summary Column* ***56***
- *Length Column* ***57***
- *Time Columns* ***57***
- *Delta (Interpacket) Time Column* ***58***
- *Absolute Time Column* ***59***

Analysis Tasks You Can Do Today **60**
- *One-Hour Baseline (Quick Look)* ***60***
- *Categorize Your Broadcast Traffic* ***60***
- *Identify Errors* ***61***
  - TCP/IP: Check out Your ICMP Traffic 61
  - IPX/SPX: IPX Burst Mode Communications 61
  - Capture Your Bootup Sequence 62
- *Capture Your Login Sequence* ***62***

Chapter 2 Test **64**

CHAPTER 3 *Packet Structures and Decodes* *66*

Those Decodes 67
How Packets are Built 68
The Application Defines the Data 68
The Protocol Stack Inserts the Transport Layer Header 69
The LAN Driver Inserts the MAC Header 71
The NIC Adds Error Detection-Spits the Bits Out 73
The Ethernet Header Formats 74
The Token Ring Header Formats 76
The IP Header Format 78
Version Field 78
Header Length Field 78
Type of Service Field 79
Total Length Field 80
Identification Field 80
Flags Field 80
Fragment Offset Field 81
Time to Live (TTL) Field 81
Protocol Field 81
Header Checksum Field 82
Source Address Field 82
Destination Address Field 82
Options Fields 83
The UDP Header Structure 84
Source Port Field 84
Well-Known Port Numbers 84
Registered Port Numbers 85
Dynamic Port Numbers 85
Destination Port Field 86
Length Field 86
Checksum Field 86
TCP Header Structure 87
Source Port Field 87
Destination Port Field 87
Sequence Number Field 87
Acknowledgment Number Field 87
Data Offset Field 88
Flags Field 88
Window Field 89
Checksum Field 89
Urgent Pointer Field 89
TCP Options Field(s) 89
Walking Through a Trace 91
The HTTP Elements 91
Uniform Resource Identifiers 92
HTTP Methods 92
Status Codes 93
Sample HTTP Communications 94
Chapter 3 Test 98

CHAPTER 4 *Identifying Typical Problems* *99*

The Elusive 'Typical Problems' 100
Ethernet/802.3 Problems 101
CSMA/CD 101 ..... *101*
Collisions (Local/Remote/Late) *103*
CRC Errors *104*
Short Frames *104*
Long Frames *104*
Alignment Errors *104*
Jabber *105*
Troubleshooting Ethernet Today *105*
Token Ring Problems 106
Token Ring 101 ..... *106*
Line Errors *108*
Internal Errors *108*
Burst Errors *109*
ARI/FCI Errors *109*
Abort Delimiter *109*
Lost Frame *109*
Receive Congestion *109*
Frame Copied Error *109*
Frequency Error *110*
Token Error *110*
Troubleshooting Token Ring Today *110*
TCP/IP Problems 111
Loving, Living, Eating and Breathing ICMP *114*
Addressing Problems *118*
Routing Problems *118*
Service Discovery Problems *118*
TCP Connection Problems *119*
IP Fragmentation Problems *119*
DHCP Problems *120*
Denial of Service Attacks *121*
IPX/SPX Problems 122
NetWare IPX/SPX 101 ..... *122*
Excessive SAP/RIP Broadcasts *124*
Routing Problems *124*
Service Discovery Problems *124*
SPX Problems *125*
Auto Framing Problems *125*
Ugly Frame Types *126*
Trace Files to Practice On 127
Chapter 4 Test 128

CHAPTER 5 *Analysis Tools and Exercises* ***131***

Laura's Lab Kit **132**
Analysis Exercises **133**
Exercise 1: EtherPeek **134**
Capturing Packets ***135***
Checking out Protocols and Packet Sizes ***136***
Opening Saved Trace Files ***138***
Exercise 2: NeoTrace Pro **140**
Checking out the Path to Podbooks.com ***141***
Exercise 3: NetScanTool Pro **143**
Using WhoIs-Learning to Gather Information ***144***
Exercise 4: PacketScrubber **147**
Scrubbing IP Addresses from a Trace File ***148***
Exercise 5: ProConvert **150**
Converting an HP Advisor Trace File ***151***
Exercise 6: Sam Spade **153**
Explore Domain Information ***155***
Exercise 7: IP SubnetCalculator **157**
Defining subnets for 191.15.0.0 ..... ***158***

Appendix A: Answers to Quizzes 161
Chapter One Answers **162**
Chapter Two Answers **165**
Chapter Three Answers **167**
Chapter Four Answers **169**

Appendix B: The Basic Flow of Data 172
The Damned OSI Model Revisited **173**
Basic Hub/Repeater Communications **174**
Basic Bridge and Switch Communications **175**
Basic Router Communications **176**
Exceptions to the Rules… **178**

Appendix C: Analysis References and Resources 179
Hardcopy Book Notes **179**
Downloaded Book Notes **179**
Multiple Book Bundle **179**
Greypapers, Whitepapers and Articles **180**
Technical Books **182**

Index **184**

CHAPTER 1

What is &#*(#&!? is Analysis?

This chapter provides an overview of analysis and its uses for troubleshooting, optimizing and testing your network. It is imperative that you have a good handle on what analysis is all about before proceeding. This is the perfect chapter to start with if you're new to network/protocol analysis.

If you are unfamiliar with how packets are routed through an internetwork, I recommend that you take a break and read Appendix B, "The Basic Flow of Data," before you venture into this chapter. - Laura

Network Analysis 101

Network analysis (also referred to as 'protocol analysis') is the art of listening in on the network communications to examine how devices communicate and determine the health of that network. Network analysis is not unlike the process a doctor uses to determine the cause of health problems. The doctor listens to your heart, lungs and complaints. Based on this information and the result of some tests, the doctor can advise you on your current health status.

During a typical network analysis session, you perform the following tasks:

Step 1: Tap in to the network.

Step 2: Capture the traffic of interest.

Step 3: View the captured traffic.

Step 4: Filter out and view just the interesting traffic.

Step 5: Document your findings.

For example, perhaps you are interested in the traffic to and from one of your HTTP servers on the network. You would follow the steps listed below:

Step 1: Connect your analyzer to the same hub the server is connected to.

Step 2: Capture all traffic to and from the server.

Step 3: View the traffic to identify any unusual packets (i.e., 100 broadcasts in succession).

Step 4: Filter out and view just the broadcast traffic from the server.

Step 5: Document the cause of the broadcasts.

Figure 1-1 shows the basic flow of data through a network analyzer. As you can see, all data that flows into the analyzer is tracked with global statistics and may trigger general alarms.

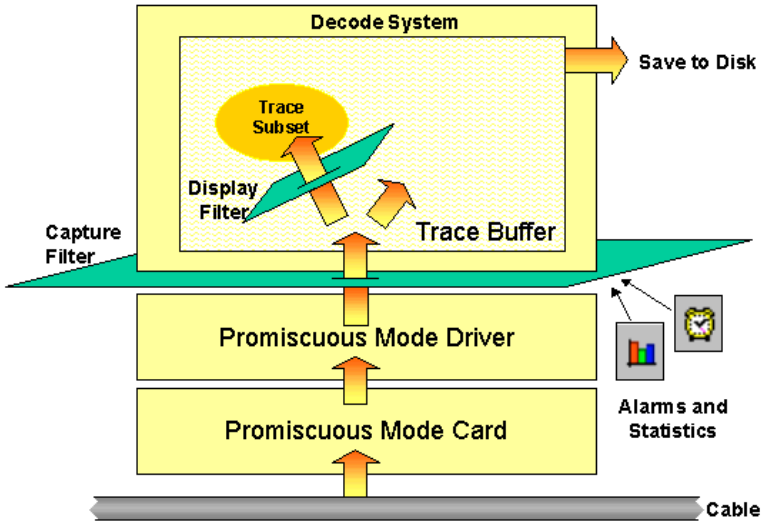


Figure 1-1: The analysis packet flow process.

A capture filter (also referred to as a pre-filter) can reduce the amount of traffic that is captured into the trace buffer (the holding place for your packets). If no capture filter is applied, all network traffic flows into your trace buffer. Yipes! On a busy network, you'll fill up the trace buffer in no time!

The display filter enables you to build subsets of the packets in the trace buffer based on some criteria. For example, if you captured all broadcast traffic on the network into the trace buffer, you may want to apply a display filter to create a subset of the trace buffer that contains IP-based broadcasts only. Even though you apply a display filter, all the packets are still in your trace buffer… you just have chosen not to view the entire contents.

In this chapter, we'll look at the various types of statistics that you can gather on your network and how to use those statistics to characterize, optimize and troubleshoot your network.

We'll examine short- and long-term trends that help you understand and work with your network's unique traffic patterns. The other patterns of interest will be the request/reply pattern structures -- using pattern analysis, you can identify good, bad and ugly applications. Finally, we'll focus on timestamping -- this includes relative, absolute and delta timestamping methods.

Alarms are definitely related to these statistics and trends, so we'll discuss alarms and alerts as well.

If your analyzer isn't shown in the screen shots of this book, don't despair -- learn everything your analyzer has to offer. If you spend the time to fully understand what your analyzer can do and you are disillusioned or frustrated, consider getting another analyzer. For those of you who stuck it out with LANalyzer for Windows --oh, come on! Get a grip... and get a more up-to-date analyzer. -- Laura

There are additional resources online at www.packet-level.com -- consult the site to download trace files, listen to audio tours of packets and communication sequences and read some of the bizarre, but informational technical materials available.

In the steps listed earlier, the capture filter would be set to grab all packets to and from the server's hardware or software address (such as the server's IP address). The display filter would then be set to filter out all broadcast traffic. The resulting filtered buffer would only show broadcast traffic from the server.

That's one example of how to use network analysis to identify a possible problem.

Network analysis is typically used for three purposes:

- troubleshooting
- optimization
- planning/testing

Troubleshooting with an Analyzer

So many possible problems can be spotted quickly and clearly on the wire once you plug in an analyzer.

For example, when one device cannot communicate with another on the network, the network analyzer can provide the ultimate answer on where the problem occurred. For example, consider a client that cannot communicate with a specific server. You can use an analyzer to discover:

- Can the client communicate onto the cabling system or is there some media fault?
- Can the server communicate onto the cabling system or is there some media fault?
- Did the service discovery process work properly? Could the client 'find' the server?
- Could the client locate the route or path to the server? Is the route/path available and within reach?
- Did the client device properly authenticate to the server environment?
- Did the client make a proper request for services?
- Did the server reply to the client's request for service?
- If the server did reply to the client, was there a failure or denial of service indicated in the reply?
- Are the client and server communicating over the same frame type?

Unfortunately, there are many communications problems that do not pipe up an error message. For example, consider that there is no error message stating that "NetBIOS is a piece of crap." <grin> -- Laura

Now let's be honest here. The world of troubleshooting can be filled with indigestion, migraines and bad backs. If you are going to maintain a career in troubleshooting and you also plan on keeping your sanity, I highly recommend that you download RFC 1925, "The Twelve Networking Truths," written by Ross Callon (self-proclaimed member of the 'Internet Order of Old Farts'). Here are the 'truths' as listed in that document:

1. It has to work.

2. No matter how hard you push and no matter what the priority, you can't increase the speed of light.

3. With sufficient thrust, pigs fly just fine, however, this is not necessarily a good idea. It is hard to be sure where they are going to land, and it could be dangerous sitting under them as they fly overhead.

4. Some things in life can never be fully appreciated nor understood unless experienced firsthand. Some things in networking can never be fully understood by someone who neither builds commercial networking equipment nor runs an operational network.

5. It is always possible to agglutinate multiple separate problems into a single complex interdependent solution. In most cases this is a bad idea.

6. It is easier to move a problem around (for example, by moving the problem to a different part of the overall network architecture) than it is to solve it.

7. It is always something.

8. It is more complicated than you think.

9. For all resources, whatever it is, you need more.

10. One size never fits all.

11. Every old idea will be proposed again with a different name and a different presentation, regardless of whether it works.

12. In protocol design, perfection has been reached not when there is nothing left to add, but when there is nothing left to take away.

Optimization

You can use an analyzer to spot all types of performance problems regardless of the protocol or media access type.

For example, consider the following optimization analysis techniques:

- Identify excess ICMP redirection messages and reconfigure devices to use a more appropriate default gateway.
- Identify and remove the cause of excessive broadcasts on the network to reduce the bandwidth overhead and processor drag caused by this traffic.
- Identify the cause of excessive 'failure' replies from a server; consider reconfiguring clients to reduce these wasteful replies.
- Tune up your router configurations to reduce the routing protocol overhead caused by excessive periodic broadcasts or multicasts.
- Remove any unsuccessful 'discovery' processes that are running wild on the network (for example, a vendor that searches for other 'same vendor' systems).
- Identify traffic patterns related to reconnaissance probes that might be performed by someone considering an attack on your network.

Every network can benefit from a quick review of the communications to identify extra unnecessary traffic.

Planning and Testing with an Analyzer

Geez…. I wish more folks would take this task seriously. Implore your management to test that ugly application before they require you to roll it out, OK?

What should you look for when you test an application or plan for growth? Very simple:

How much bandwidth does a single user require to run the application on the network? Multiply this number by the number of users you'll eventually supply the application to. Do you have enough bandwidth on the segment to support that application?

In one case, a company had planned on rolling out a corporate screensaver to hundreds of desktops across the WAN. Through testing, it became evident that if the screensaver were rolled out, it would consume the entire bandwidth of the WAN links, leaving no room for actual data to cross the network. -Laura

How do you test an application? Consider Figure 1-2. Simply plug your analyzer into a port that enables you to see the communications between your client-server test devices. (Yes, application testing should be done in a lab, not on the live network.)

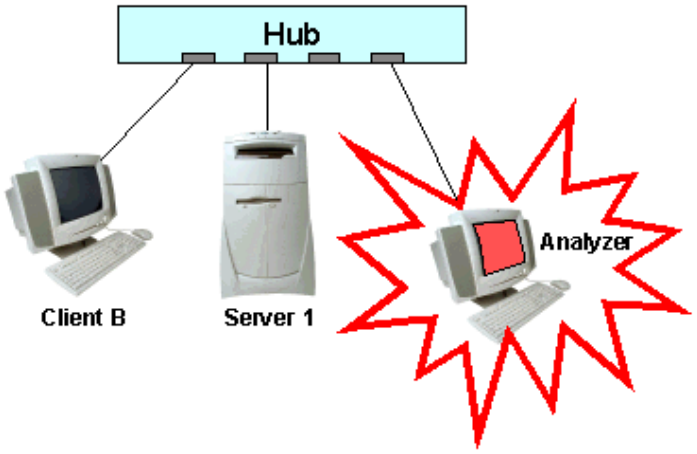


Figure 1-2: Application analysis setup.

Track each procedure of the application. For example, the initial launch sequence, the login sequence, the basic operations, and the close-down sequence.

For more information on application testing, refer to "Advanced Network Analysis Techniques." I went through packet-by-packet analysis of an FTP file transfer and HTTP web browsing process in Chapter 4, "Manual Decoding." Also check out the April 1999 issue of NetWare Connections Magazine (www.nwconnection.com). That article details a typical onsite visit (including an application test). -- Laura

What the Heck is a Network Analyzer?

A network analyzer is a device (desktop, laptop or portable computer) that can 'capture' all the packets seen on the network and display them in the order they appeared on the cable. A good analyzer should have some alerts/alarms that notify you of unusual or faulty traffic patterns. The analyzer should also be able to build trend graphs to illustrate the current and long-term traffic patterns (such as utilization and packets per second).

In order to make the communications information useful to you, the analyzer decodes, or interprets, the actual packet information received. Each analyzer has a different set of decodes built by their engineers (or perhaps outside contractors).

One example of a decode is the value displayed in the type field of an Ethernet header. In the example shown in Figure 1-3, you can see the word IP next to the hex value 0x0800. The leading '0x' denotes this is a hexadecimal value.

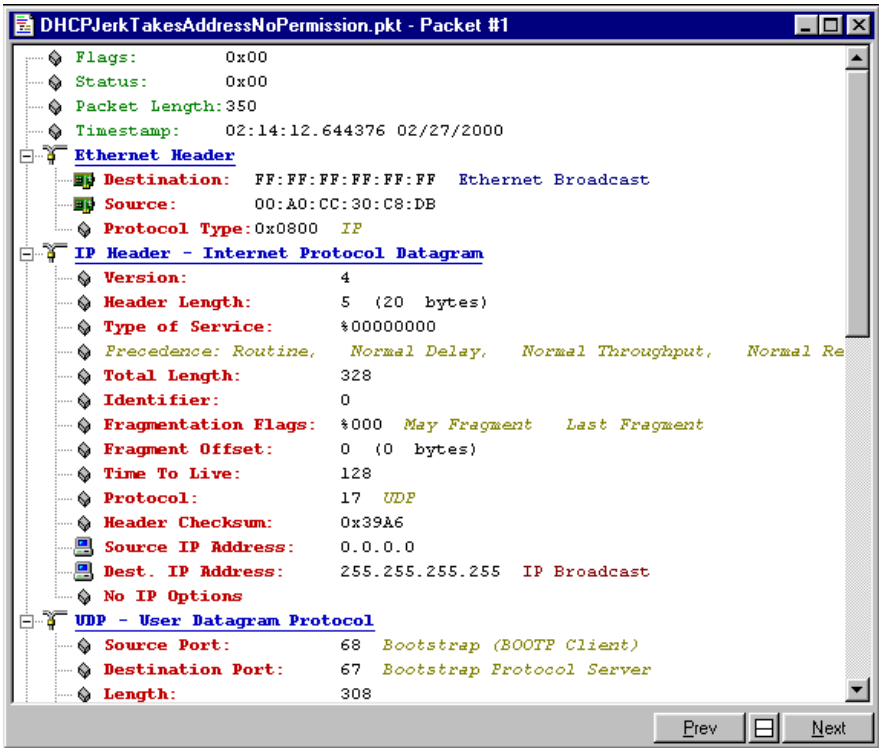


Figure 1-3: The decode interprets the value 0x0800 as 'IP.' [EtherPeek]

By capturing the packets and viewing the 'conversation' between devices, you can identify communication patterns, spot errors and check the efficiency of these communications.

For example, consider a client that cannot connect to a server. By capturing and viewing the bootup traffic, you can identify where the problem lies. You can systematically rule out elements such as cable faults, card faults, protocol faults, configuration faults, service discovery faults, route location faults, authentication faults, etc.

The trend information provided by an analyzer can help you determine how much bandwidth (percentage of total bits per second that can be transmitted) and packets per second are being used by an application or group of devices. This information can help you troubleshoot a congested network and determine the minimum capabilities required of interconnecting devices.

The term 'protocol analyzer' and 'network analyzer' are interchangeable. Years ago, marketing pukes, uh... er.... 'folks' started using the term 'network analyzer' because some buyers are intimidated by the term 'protocol analyzer.' --Laura

Refer back to Figure 1-1 again. Although that is a very simplified view of a network analyzer, it covers the basic functionality.

In order for these analyzers to capture ALL the traffic, they must have a network interface card and driver that supports 'promiscuous mode' operations. A promiscuous mode card is able to capture error packets and packets that are not addressed to the local card. Of course, broadcast and multicast traffic should also be visible to the analyzer.

Most current analyzers require NDIS version 3.0+ drivers to support promiscuous mode operations. In some cases, the analyzer manufacturer offers a specialized driver to enhance the card and provide error reporting. This does not mean that all cards can run in promiscuous mode -- some cards use special functionality to enhance performance at the expense of promiscuous mode operations. For example, The EtherLink III MLID (3C589.COM) supports promiscuous mode operation. However, the adapter will not report bad packets. The Parallel Tasking technology incorporated into the EtherLink III family of adapters increases network performance by discarding bad packets. If you must capture bad packets, don't use an EtherLink III adapter.

You only need a promiscuous mode card on the actual machine that is capturing the packets -no other devices in your network need promiscuous mode functionality.

Standalone v. Distributed Analyzers

There are two types of network analyzers -- standalone analyzers and distributed analyzers.

- Standalone analyzers only capture packets that cross the wire that the network the analyzer is connected to.

- Distributed analyzers can capture packets on remote networks through the use of probes (or remote agents) that capture the packets on another network and send them to a centralized analyzer station for viewing.

You need to purchase and implement an analyzer solution based on how your network is designed. A hubbed network requires a different analyzer than a heavily routed network. And, of course, switched networks provide a whole new challenge.

How to Use a Standalone Analyzer

Standalone analyzers are the most popular type of analyzer. You just take your portable and plug in where you want. You could also set up your own desktop machine as a standalone analyzer and monitor all visible traffic from the comfort of your own desk.

I use standalone analyzers almost exclusively. This is because I travel to so many companies and connect to so many different networks. If I were an 'in house analyst' at a large company, I'd probably seriously consider a distributed analyzer solution. First, however, I'd do some research on the traffic that flows between the remote analyzer and the analyzer manager station. --Laura

Standalone analyzers can only capture data that is seen locally. For example, if your analyzer system is connected into a hub with four other devices (router, two PCs and a server), as shown in Figure 1-4, it can see all traffic that flows through that hub. The analyzer cannot see traffic that is on the other side of the router, however.

In order to view traffic on the other side of the router, the analyzer would need to be moved to a hub port on the other side of the router. Refer to the “Analyzer Placement” section next for more information on analyzer visibility.

This is a common way to perform analysis. In fact, much of my onsite life is spent in wiring closets moving my analyzer from one port to another. One suggestion - store a folding chair and table in those wiring closets - those floors are damn cold and hard. -Laura

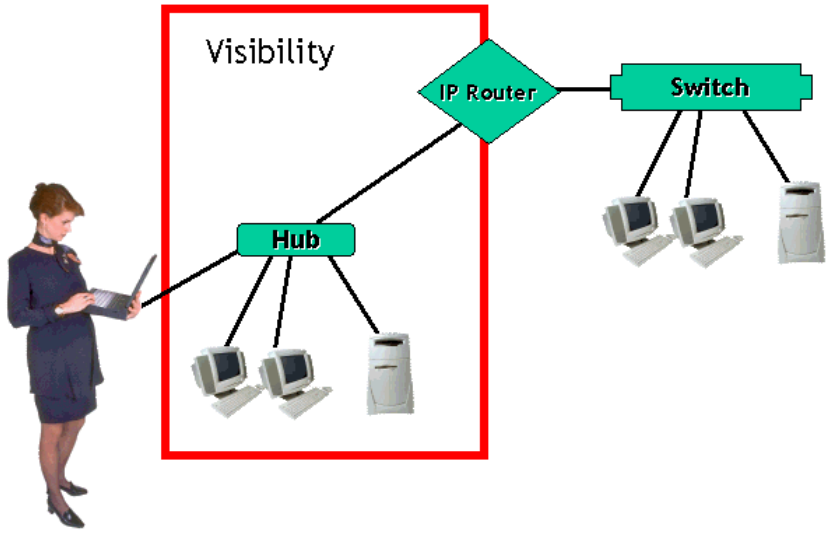


Figure 1.4: What can a standalone analyzer "see?"

How to Use a Distributed Analyzer

Distributed analyzers (such as Distributed Sniffer), can deploy probes or remote agents that capture traffic on other networks. The captured traffic is then sent to the manager device for analysis.

Figure 1-5 shows an example of what a distributed analyzer can 'see' based on the configuration. Since we've placed an analyzer agent on the router, we can capture data on both sides of the router. We've also placed an agent on the switch, enabling us to see all traffic that travels through that switch.

On a larger network, a distributed analysis solution can offer you the ability to troubleshoot multiple buildings or sites from a single location. Of course, you should examine the traffic that is crossing the network to ensure that you are not causing problems (such as adding extra load onto the network).

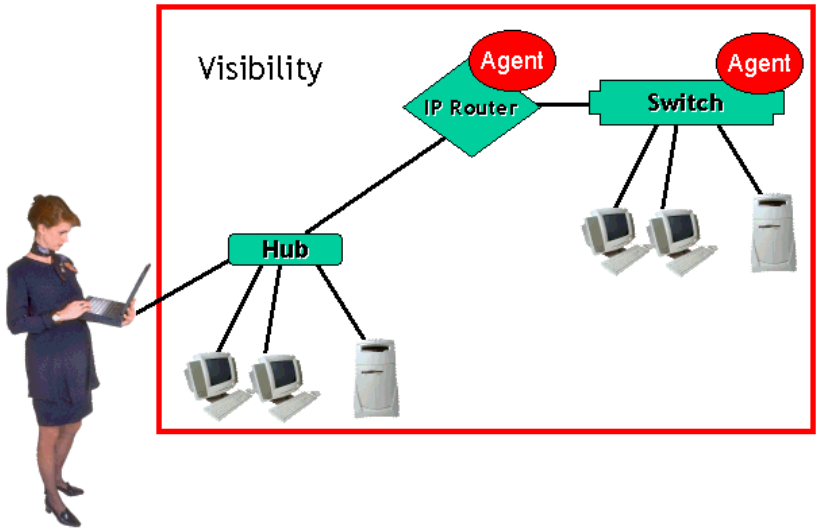


Figure 1-5: The distributed analyzer 'view.'

One of my clients has a wonderful 'War Room.' They have set up their distributed analyzer 'map' screen (that depicts the entire network in graphical format) to display in an 8' x 8' projection on a wall. When you enter the room, the map provides an immediate visual on the status of the entire network. They have also added some audible alarms that 'announce' the result of SNMP alert messages as they are received at the management console. Interestingly, some alerts have been interpreted and recorded into songs, jokes, bodily sounds, and other unmentionable noises. Lots of fun to visit. -- Laura

Check with your hub, switch or router vendor to see what distributed solution they suggest, if any.

Software v. Hardware/Software Analyzers

OK, so now you're ready to purchase an analyzer - you've selected between a standalone and distributed solution. What's next? Well… are you interested in an analysis solution that is software only or would you consider a hardware/software solution?

Software solutions (such as EtherPeek and Sniffer Basic/Pro) are typically less expensive than hardware/software solutions. They are also generally easy to install (provided there are no software conflicts and you have a decent promiscuous mode card).

Hardware/software solutions are typically more expensive, but they typically offer greater functionality for this expense.

If you are just starting out in analysis, I recommend that you work with the less-expensive solutions, such as EtherPeek or Sniffer Basic. You can get the demo version of EtherPeek online at www.wildpackets.com. *At the time of this writing, Network Associates is not so open with their demo version, unfortunately. Check* www.sniffer.com. *- Laura*

Analyzer Placement

OK… now where do you place this analyzer solution? Well, that depends entirely on the design of your network. Check each of the sections below to learn how interconnecting devices such as hubs, bridges, switches and routers affect the network traffic and how to analyze each design.

Hubbed Network

Easy peasy! This is the simplest network to hook up an analyzer to because all traffic goes everywhere. (By the way, that traffic pattern is also the reason that you desperately need an analyzer.) Just plug the analyzer into any hub to view all traffic from all devices, as shown in Figure 1-6.

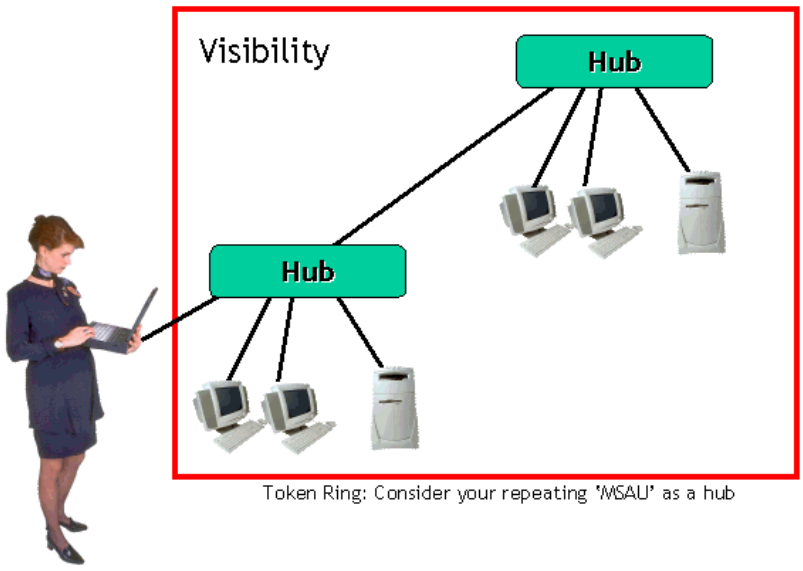


Figure 1-6: You can see everything on a hubbed network!

Bridged Network

It would be difficult to go out and purchase a bridge today. They have been replaced by switches. This section is for those of you who have already installed bridges and need to analyze your traffic.

Consider the purpose of a bridge - a bridge should isolate and localize traffic based on hardware addresses. As shown in Figure 1.7, if a packet from Client A is destined for Client B and both devices are on

the same side of the bridge, that bridge will not forward the packet. If Client A sends a packet to Client C, however, the bridge will forward the packet to Client C's segment because the bridge knows that Client C is located off port 2.

If Client A sends a broadcast, the bridge will forward the packet (after all, it was destined for all devices). If Client A sends a packet to an unknown address, the bridge will forward the packet - bridges always forward packets that they do not have an entry for.

In Figure 1-7, we have placed analyzers on both sides of the bridge to ensure that we can get the traffic from each side. That solution (or perhaps a distributed analyzer solution) will help you analyze bridged traffic.

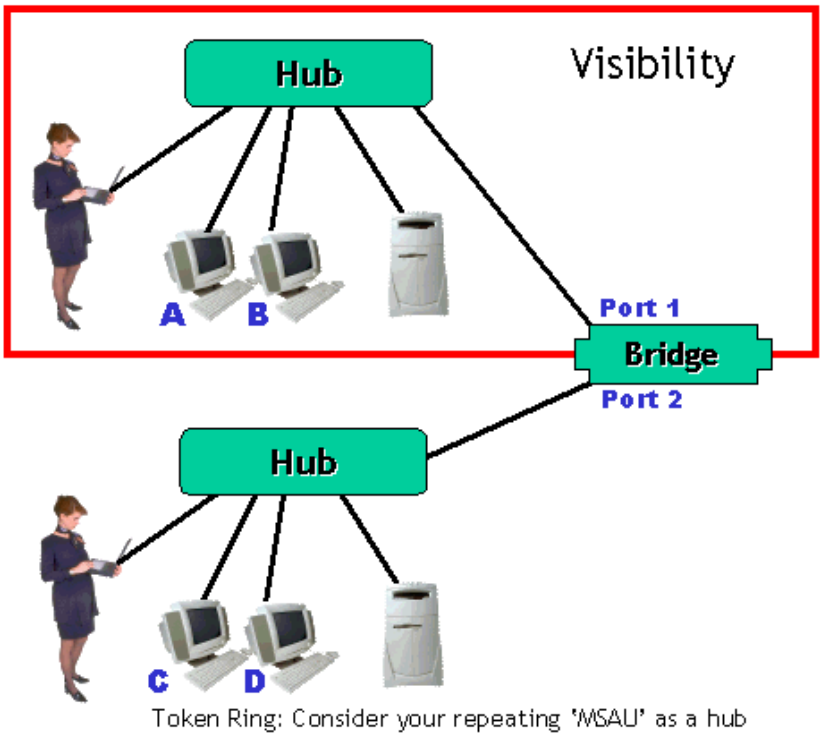


Figure 1-7: Bridged network.

Switched Network

So you bought a switch to help control and isolate network traffic, thereby allowing more efficient use of the bandwidth. Aaaaaargh! This is a great technique for reducing unnecessary traffic on connected ports, but it creates anguish to the protocol analyst.

Try it out! Plug your analyzer into a switch port and classify the traffic. You'll only see a bunch of broadcast and multicast packets. Of course, you should also see packets specifically addressed to your analyzer's card address if there are any.

Traffic from one device on a switch only flows directly to the destination device on another port. In Figure 1-8, the client and server share a virtual circuit between them. Their traffic flows from Port 1 to Port 3 and vice versa. Their traffic is not sent down any other ports. If you plug your analyzer into the switch, you won't be able to 'eavesdrop' into the communications because the switch is doing what it should be doing - it is isolating local conversations.

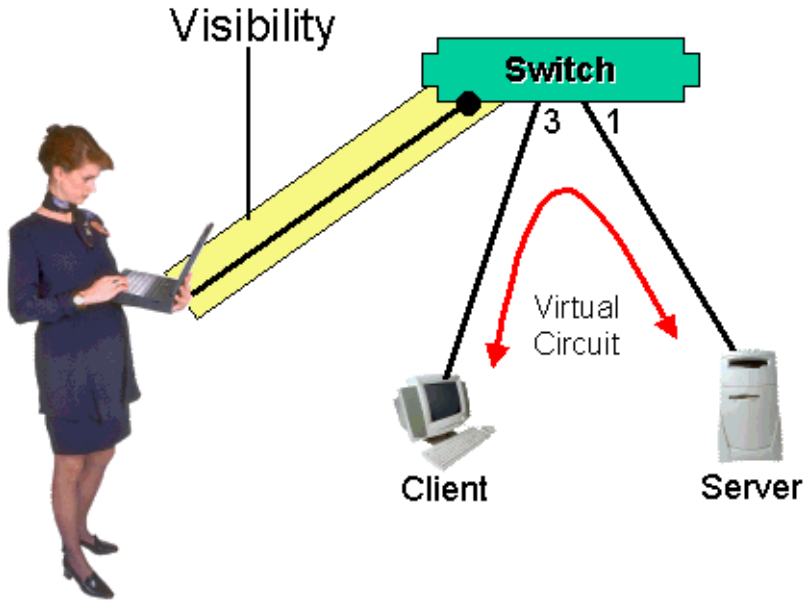


Figure 1-8: A switched communication creates a 'virtual circuit' between devices.

There are two basic types of switches -- layer 2 switches (pure switches that act like bridges) and layer 3 switches (switches that are 'cross-dressing' as routers these days by providing layer 3 routing services whenever possible). The lines separating the functionality of the two devices have blurred. - Laura

It is because of this natural traffic isolation process that you will have problems seeing traffic other than broadcasts/multicasts and your own traffic. So, consider the following ways to analyze a switched network:

- Hub out
- Deploy analyzer agents
- Configure switch for port spanning or mirroring

Let's take a closer look at each of these options.

Hubbing Out

This is really cheating, but it works great when you need to see traffic to and from some device that is connected to a switch.

Simply place a hub between the device of interest (a server, for example) and pop your analyzer into one of the hub ports, as shown in Figure 1-9.

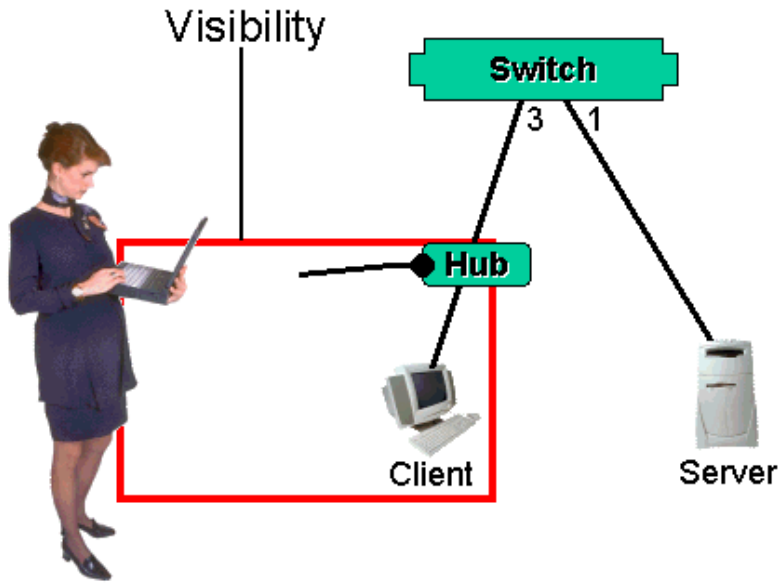


Figure 1-9: Hubbing out to an analyzer.

You must either connect the switch to the hub's crossover port or use a crossover cable to connect the two.

Now, true, this is a bit of a cheat - but it works great. Only problem is that you must move the hub/analyzer combination around from

device to device if you want to look at multiple devices around the network. A distributed analysis solution (covered next) may be more practical for you.

This solution works fine on standard half-duplex Ethernet networks. If you use full-duplex connections, however, you'll need to install a full-duplex tap. See the Full-Duplex Analysis section later in this chapter. -- Laura

Analyzer Agents

Analyzer agents are used by distributed analyzers. These agents are typically software programs that are loaded on switches to enable them to capture traffic from all ports and send the data to a management console.

The beauty of this is that you can manage switched traffic from a central location. Unfortunately, however, you might get caught up in a proprietary solution or find that this type of feature makes the switch too expensive.

Port Spanning/Mirroring

This is my favorite way of dealing with switched networks, but can only be used if the switch supports this functionality. Simply configure the switch to send a copy of any port's traffic down another port - the one your analyzer is connected to. Some vendors call this port spanning (SPAN stands for switched port analysis); others call it port mirroring. (Cisco uses the term port spanning, whereas Bay/Nortel uses the term port mirroring.)

As shown in Figure 1-10, the traffic from ports 1 and 3 is copied down to port 7 which is where the analyzer is located.

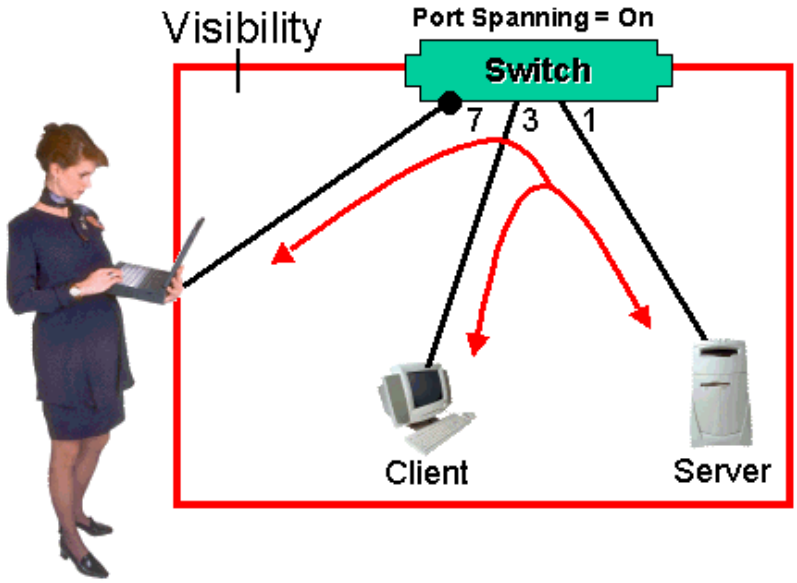


Figure 1-10: Spanning ports enables packet copies to be sent to the analyzer's port.

The following is a sample configuration to set up switch port spanning on a Cisco Catalyst 2900XL:

```
Switch(config)#int fa0/7
Switch(config-if)#port monitor fastEthernet 0/1
Switch(config-if)#port monitor fastEthernet 0/3
```

Routed Network

Routers isolate traffic based on the network (software) address. If you place an analyzer on one side of a router, you should only see traffic that is destined to that network. For example, in Figure 1-11, we have two networks (10.1.0.0 and 10.2.0.0, subnetted 255.255.0.0). Traffic between the clients and servers on network 10.1.0.0 will not be visible to the analyzer on network 10.2.0.0.

In this case, we have three options:

- Place a standalone analyzer on each side of the router.
- Load an analyzer agent on the router (make sure the analyzer agent is a multisegment agent that can capture packets from both connected networks).

- Load an analyzer agent on devices (such as the file servers) on each side of the router.

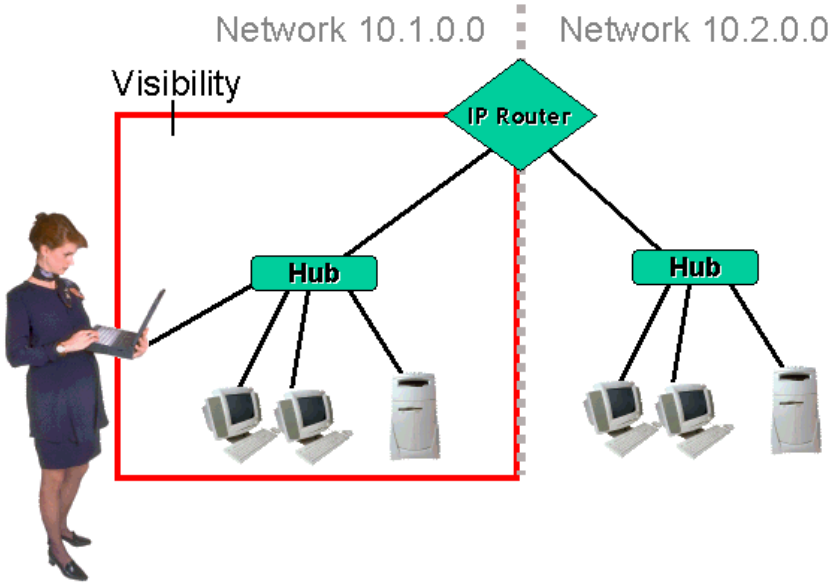


Figure 1-11: The standalone analyzer can only see the traffic on network 10.1.0.0.

WAN Link

WAN links typically consist of routing devices at each end and a link connecting them. Treat this link just as you would a routed network - place an analyzer or analyzer agent on each side of the WAN routers.

Ugly WAN links consist of bridges on each side - Yuck. Bridges forward all broadcasts and multicasts by default. That traffic is highly undesirable across an expensive bandwidth-starved WAN link. Please use a router on each side of your WAN link. If you are WAN bridging to support an unroutable protocol, consider encapsulating the protocol inside a routable one (such as IP). This enables you to set up routers across the WAN link. Data Link Switching (DLSw) is an example of this nice encapsulation technique.-Laura

---

How do you place an analyzer on the WAN link? Well, it actually depends on the WAN link and the analyzer solution. Figure 1-12 and 1-13 illustrates two possible solutions using Network Associates Sniffer Internetwork Analyzer.

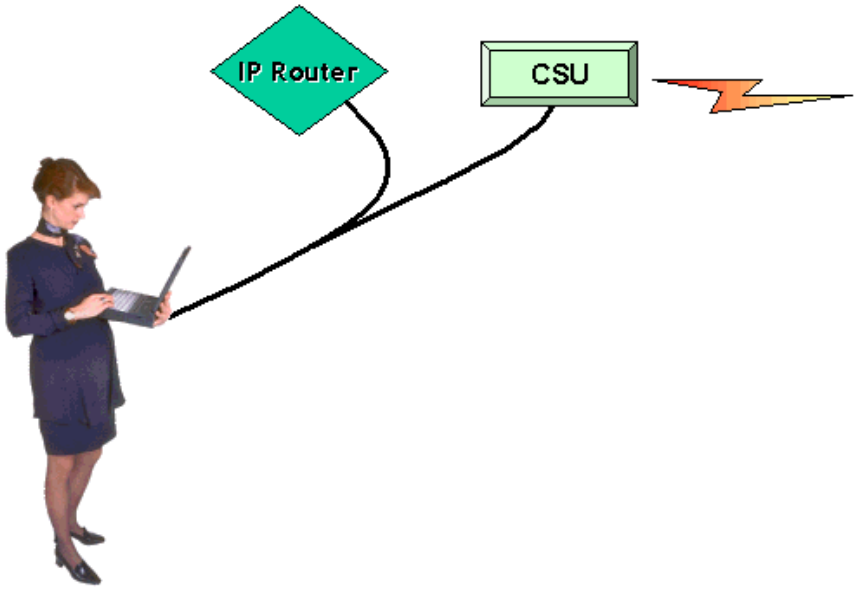


Figure 1-12: Connect the WAN analyzer to the router and CSU using a Y-cable.

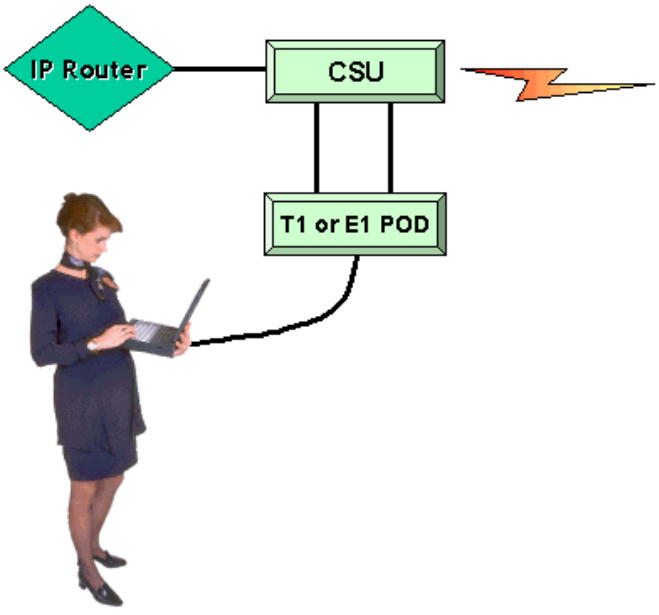


Figure 1-13: Connect the WAN analyzer to the CSU monitor port.

As you can see in Figures 1-12 and 1-13, you can either install the Sniffer Internetwork Analyzer through the use of a Y-cable between

the router and the CSU or through a connection to the CSU monitor port.

Your WAN analysis vendor can provide additional details on how to make the physical connection for your WAN link.

Full-Duplex Connections

Full-duplex communications in effect, turn a one-lane road into a two-lane road with one transmit channel and one receive channel. In order to view the traffic traveling on both channels, you need a full-duplex tap of some sort. The simplest one I know of is the Century Tap from Shomiti.

As shown in Figure 1-14, full-duplex taps combine the transmit (TX) and receive (RX) data stream into one RX stream that is sent down to the analyzer.

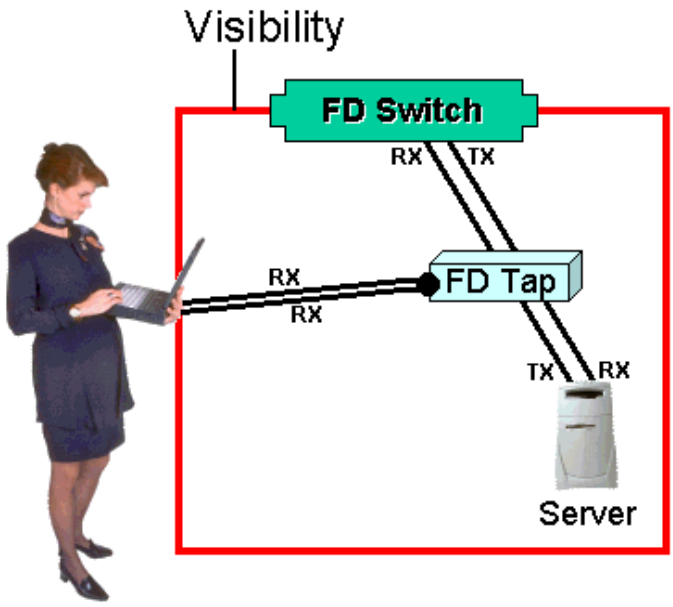


Figure 1-14: A full-duplex tap combines the data stream and sends it to the analyzer.

So…. as you can see, the traffic isolating nature of the devices on your network will govern where you should place your analyzer. It's really quite simple. Place the analyzer (or analyzer agent) on each side of any device that might filter traffic.

In the next section, we'll examine the elements of a network analyzer. You will use these elements in Chapter 2 to analyzer your network today!

Analyzer Elements

Every analyzer has a different set of elements. Some analyzers share a set of elements, but call them by different names. For example, some analyzers use the term "capture filter" when others use the term "pre-filter." They do the same thing and are listed together in this chapter.

Consider the picture of analyzer functionality and elements shown earlier and repeated in Figure 1.15.

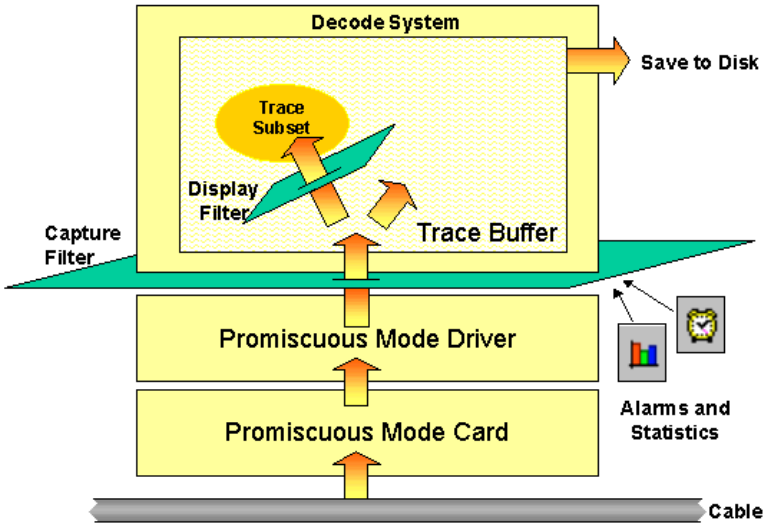


Figure 1.15: The flow of data through an analyzer.

Figure 1.15 depicts the following analyzer elements:

- Promiscuous Mode Card and Driver (where packets flow into the analyzer)
- Decodes (wrapped around all packets captured)
- Capture Filters (defining packets that should be placed in the trace buffer)
- Display Filters (creating a subset of packets in the trace buffer)
- Alarms/alarm thresholds (alerting you to possible problems)
- Trace buffer (holding all packets captured)

Let's check out these elements and what they can do for you.

Promiscuous Mode Card and Driver

Naturally, the type of card you use dictates the media that you can analyze. If your analyzer has a 10Base-T card, then you can only connect to a 10Base-T network, not a token ring network (duh). We covered the specifics about promiscuous mode operations earlier in this chapter, so I won't repeat myself.

Decodes

The decodes control how packet contents are displayed on the screen. They are the 'deciphered' version of the bits on the wire. For example, Figure 1.16 shows a decoded DHCP (Dynamic Host Configuration Protocol) packet. If the analyzer didn't have a decode for that protocol, you'd see just a big hexadecimal 'dump.'

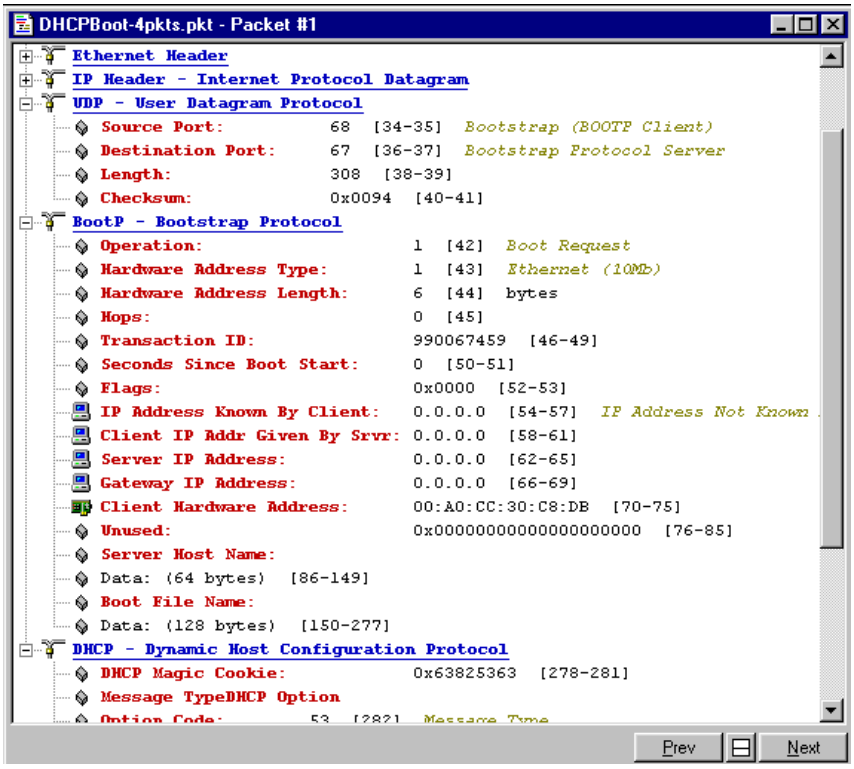

DHCPBoot-4pkts.pkt - Packet #1
Ethernet Header
IP Header - Internet Protocol Datagram
UDP - User Datagram Protocol
Source Port: 68 [34-35] Bootstrap (BOOTP Client)
Destination Port: 67 [36-37] Bootstrap Protocol Server
Length: 308 [38-39]
Checksum: 0x0094 [40-41]
BootP - Bootstrap Protocol
Operation: 1 [42] Boot Request
Hardware Address Type: 1 [43] Ethernet (10Mb)
Hardware Address Length: 6 [44] bytes
Hops: 0 [45]
Transaction ID: 990067459 [46-49]
Seconds Since Boot Start: 0 [50-51]
Flags: 0x0000 [52-53]
IP Address Known By Client: 0.0.0.0 [54-57] IP Address Not Known
Client IP Addr Given By Srvr: 0.0.0.0 [58-61]
Server IP Address: 0.0.0.0 [62-65]
Gateway IP Address: 0.0.0.0 [66-69]
Client Hardware Address: 00:A0:CC:30:C8:DB [70-75]
Unused: 0x00000000000000000000 [76-85]
Server Host Name:
Data: (64 bytes) [86-149]
Boot File Name:
Data: (128 bytes) [150-277]
DHCP - Dynamic Host Configuration Protocol
DHCP Magic Cookie: 0x63825363 [278-281]
Message TypeDHCP Option
Prev
Next

Figure 1.16: Decoded DHCP packet. [EtherPeek]

When an analyzer decodes a packet for you, it breaks down the packet contents and lists the individual fields, the field contents, and their meanings. For example, the decode in Figure 1.16 breaks down the packet through the entire DHCP options to list the magic cookie, message type, client IP address values and more.

If a DHCP decode was not available with EtherPeek, you would need to look up the field lengths and values in the DHCP specifications (RFC 2131) and manually decode the packets.

Not all analyzers have quality decodes. It is always a good idea to have access to resources that would enable you to manually decode a field if required. Check out the end of this chapter and Appendix C, "Resource List" to find out where you can get packet and protocol specifications. --Laura

Capture Filter (Pre-Filters)

We've already discussed capture filters a bit in this chapter. These filters define the traffic that should be copied from the network into the trace buffer. For example, you may set a capture filter for traffic to and from Fred's workstation. Or perhaps you are interested in all IPX/SPX traffic only.

If you've ever seen my lectures, you've heard about the infamous 'Fred, User From Hell (UFH)." Every network has one. - Laura

By setting a capture filter, you reduce the number of packets actually captured to focus on the interesting traffic.

Display Filters (Post-Filters)

Display filters create subsets of the trace buffer. Display filters allow you to reduce the number of packets that you need to wade through to a manageable number (hopefully). For example, if you set a capture filter for all IPX/SPX traffic and then wanted to determine what types of broadcasts occurred on the network, you would apply a display filter looking for only packets addressed to the broadcast address (0xFF-FF-FF-FF-FF-FF).

What types of traffic should you look for specifically? On TCP/IP networks, you want to filter out all broadcasts and ICMP traffic. On IPX/SPX networks, you want to filter out all broadcasts and NCP failure packets. We'll work on this more in Chapter 2.

Gauges/Graphs

Gauges and graphs can help you by giving you a graphical view of the network health and traffic flow. It really is true that a picture is worth a thousand words… or in this case, a thousand packets.

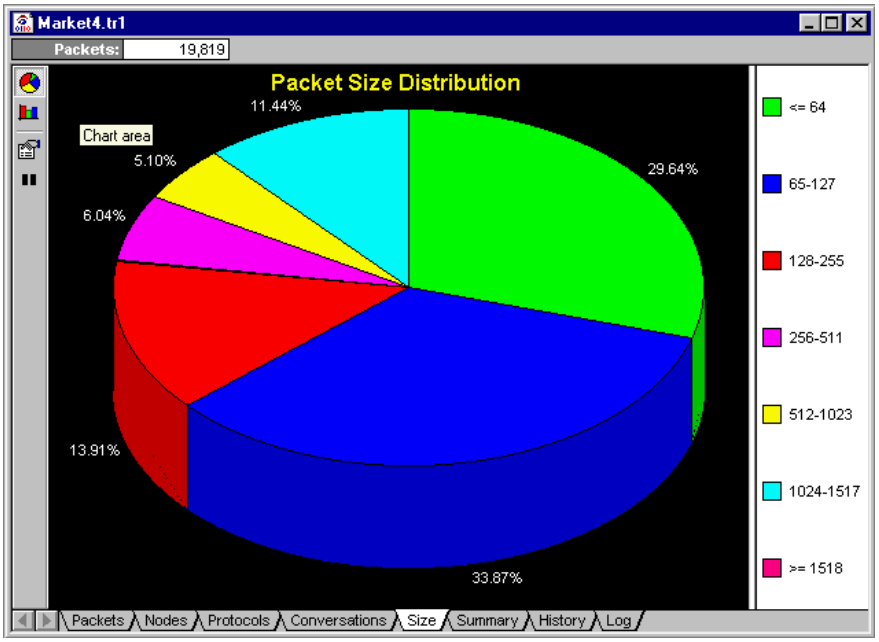


Figure 1.17: A picture is worth a thousand packets. [EtherPeek]

Figure 1.17 shows the packet size distribution chart on the EtherPeek. This chart indicates that almost 30% of this network's traffic is minimum sized (64 bytes). Not very efficient... too many packet headers to carry the data.

Be certain to include as many graphical references as possible when you are creating a report that documents your network health. They enable the reader to 'see' your points much faster and more clearly. We'll cover report generation and format techniques in Chapter 2.

Alarms/Alarm Thresholds

There is no way that you can just peruse through 10,000 packets and catch all the network problems instantly. This is where the analyzer's alarms become invaluable. There are alarms that alert you to data link errors, network congestion, too many packets per second, slow applications, application faults, routing problems, service discovery faults, address allocation errors, etc.

Figure 1-18 shows a sample of alarm threshold settings for the EtherPeek.

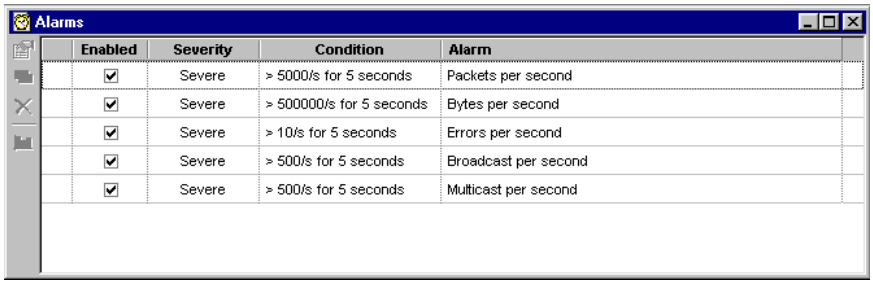


Figure 1-18: The basic EtherPeek alarm list.

Which alarms should you really watch? Well… here's a bunch that I like to keep close watch of:

- **Utilization percentage**: shown as a percentage of the bandwidth (available roadway) that is used up. On Ethernet networks especially, the performance degrades significantly when the utilization gets up there above 40%. Watch the collision/fragment error count in relationship to utilization. Chapter 4, "Identifying Typical Problems," covers this relationship in greater depth.
- **Packets per second**: shown in number of packets seen per second on the network. This number can give you an idea of how many packets an interconnecting device (such as a router or switch) will need to process per second.
- **Broadcasts per second**: Yuck. This number tells you how much blabber is on your network. Broadcasts (packets addressed to 0xFF-FF-FF-FF-FF-FF) need to be processed by all devices (regardless of their operating system or protocol). Excessive broadcasts will 'hose' your network. Chapter 4, "Identifying Typical Problems," provides more detail on broadcast storms.
- **Server/Router down**: Duh.
- **MAC-layer errors**: shown in 'per second' increments. These MAC (media access control) layer errors are defined as layer 1 and 2 errors that corrupt packet formats or make access to the network impossible. Check out Chapter 4, "Identifying Typical Problems" for more details on this one.

Those are just the basic alarms to watch out for. Each analyzer product has a different set of alarms.

Trace Buffer

The trace buffer is the storage location for all the packets you capture - it is the true 'bit bucket.' Typically, the trace buffer is a reserved area of memory in the local system and is, therefore, limited in size to the amount of memory in the machine.

Strangely, people think 'the bigger, the better' on the trace buffer size. Try to remember however, that capturing 1 million packets is not the most effective methodology - you should try to define an area of interest to reduce the number of packets to wade through. Of course, if you have no life, 1 million packets may be great company on a Saturday night. -Laura

When you apply a display filter on the contents of the trace buffer, you do not override the contents of the original buffer - you simply are viewing a subset of the full buffer.

Most analyzers have their trace buffers set up for cyclical operations by default. This simply means that the analyzer will keep capturing packets even after the buffer has filled up completely. The analyzer will dump the oldest packets captured to make room for the newer ones - FIFO (first in, first out).

The more sophisticated analyzers have triggers that define when specific actions take place (such as starting or stopping the capturing process or launching a supplemental application). Although you might not use this functionality much, in many instances you can benefit greatly from these triggering capabilities. For example, perhaps you set a trigger to alert you via beeper when a 'server down' alarm is generated on the analyzer. Nice, eh?

Check to see if your analyzer offers triggers. Try a simple trigger test - set up your analyzer to capture all traffic, but stop when the broadcast rate exceeds 100 broadcasts/second. Now, I don't actually know what your allowable broadcast rate should be on your network without seeing more information on your traffic characterization, but 100 broadcasts per second is a good starting threshold. -Laura

In the next section, I'll divulge some of my secrets for keeping up with the analysis world and where you can go for more information on any of this stuff.

Your Basic Online Resource List

This section is one you should come back to when you are by your computer. You'll want to bookmark the URLs listed in this section. Make sure you also check out Appendix C, "Resources and References."

Protocol Analysis Institute - www.packet-level.com

Yup - your #1 resource is the Protocol Analysis Institute web site - where we are working on keeping my reference materials and links up-to-date so you can go to one location for all sorts of goodies. You can get supplemental files for this podbook or any other trace files that look interesting.

If there's something you want added to the reference section online, please let me know via email at resources@packet-level.com.

Internet Assigned Numbers Authority - www.iana.org

These are the TCP/IP numbers folks. They assign and document all the field values that you see inside those packets. If your analyzer doesn't have a decode for a field, check out this site to lookup the number values assigned. We keep a copy of some of the interesting number lists over on www.packet-level.com as well.

Institute of Electrical and Electronic Engineers - www.ieee.org

This is the place to go to get your 802.3 (Carrier Sense Multiple Access/Collision Detection), 802.5 (Token Passing Ring), and 802.11 (Wireless LAN) specifications. If you're an IEEE member, you'll receive discounts on your specification purchases.

Internet Engineering Task Force - www.ietf.org

These are the folks that work diligently on the Internet specifications that we have all come to know and love over the years. These are the geeks of the geeks, the nerds of the nerds - many of whom are probably millionaires by now. If you want to know how the specifications are coming along, join their mailing list and stand back! The email can be overwhelming.

You can find more references and additional resources online in www.packet-level.com's reference section.

Chapter 1 Test

OK, suck it up folks… step right up and take the test to determine if you were awake when you read this chapter.

The answers are in Appendix A - but don't cheat - really give it a good try before looking up the answers. -Laura

1.1 How do you reduce the number of packets that are captured into the trace buffer?

_____

1.2 What is the purpose of the display filter?

_____

1.3 Where is the trace buffer located?

_____

1.4 What is the purpose of a decode?

_____

1.5 Why do you need a promiscuous mode card on your network analyzer?

_____

1.6 What are the primary elements of an analyzer?

_____

1.7 What are the three ways of analyzing a switched network?

_____

1.8 What format are communications shown in when an analyzer does not have a decode for the protocol?

_____

1.9 What is the value of the basic broadcast address?

_____

1.10 What is a cyclical trace buffer? How would you use it for troubleshooting?

_____

CHAPTER 2

How Do You Start?

This chapter will ramp you up to start capturing packets, manipulating those packets and getting the feel for performing efficient analysis. A number of exercises sprinkled throughout should give you a chance to try some basic through advanced tasks. If you still feel a bit wobbly on the whole communications stuff, check out Appendix B, "The Basic Flow of Data," which provides a nice primer on how packets travel from your local machine through a router to a device on another network.

Go ahead and start up your analyzer now - if you don't have an analyzer yet, go check out www.wildpackets.com *-- they have a great demo of EtherPeek and TokenPeek online. Chapter 5 contains a lab test list that shows you how to use EtherPeek to analyze your network. I also wrote a document entitled, "10 Cool Things You Can Do With the Free EtherPeek Demo." -Laura*

Overview of the Process

OK, here we go! In this section, I'm going to take you through the steps that I follow when I go onsite to a customer. You learn how to get a general view of the network and 'drill down' to the important areas that indicate possible problems. Finally, you learn how to create a simple, concise document that explains your findings.

In general, a network analysis session consists of the following steps:

1. Identify areas of interest.
2. Tap in at an appropriate point (and optionally, as in the case of a switched network redirect traffic to the tap in location).
3. Perform the 'laying on of hands' while capturing all traffic.
4. Begin to classify the traffic and document trends.
5. Identify possible 'hot spots' (problem areas).
6. Filter the trace buffer to focus in on specific conversations and packet types.
7. Document your findings.

That's all! It may appear daunting, but it's really not. Let's take a closer look at each of the steps and we can really concentrate on the technique of the 'laying on of hands.' Chapter 4 in this book provides additional details on how to identify hot spots and where problems may arise on typical networks.

Identify Areas of Interest

This should be the easiest part of analysis. Simply put, what do you want to know about? Are you interested in the backbone traffic? What about the traffic on a specific segment or perhaps a certain department's traffic? Maybe you want to know what the traffic looks like on the WAN link or the link connecting to a new site. How about the traffic to and from Fred's workstation or the critical router on the network?

Even if you have a distributed analyzer, you still need to select the area of the network that you want to analyze.

Why not start with the network that you are connected to?

Keep in mind that you must have a promiscuous mode driver and card to perform analysis. If you are connected to a switch, you also need to configure the switch for port spanning or mirroring. Refer back to Chapter 1 for more information on promiscuous mode analysis and switched network analysis. -Laura

Tap In (and Optionally Redirect Traffic)

So now you've decided what network is of interest to you. It's time to connect your analyzer to the network. Where you 'tap in' depends on the type of network you have. Remember that the following devices will isolate packets to control traffic more efficiently. These devices include:

- Bridges
- Switches
- Routers

Although you can simply move your analyzer from one side of a bridge or router to the other side to see filtered traffic, a switch requires more work. Hopefully the switch supports port mirroring or spanning. Refer back to Chapter 1 for a quick review of analyzer placement options.

Perform the 'Laying on of Hands'

The 'Laying on of Hands" is the first step in analyzing the performance of a network. It is, quite literally, the process of looking at all packets on the network. It sounds tough, but it really isn't too bad -- you may need to practice a few times to get the hang of it.

To start, set up your analyzer without any capture filter and capture at least 8 MB of packets.

You may need to reconfigure your analyzer trace buffer size to enable you to capture 8 MB of packets - sometimes the default buffer size is much smaller. Figures 2-1 and 2-2 shows the configuration steps for both EtherPeek and Sniffer Pro.

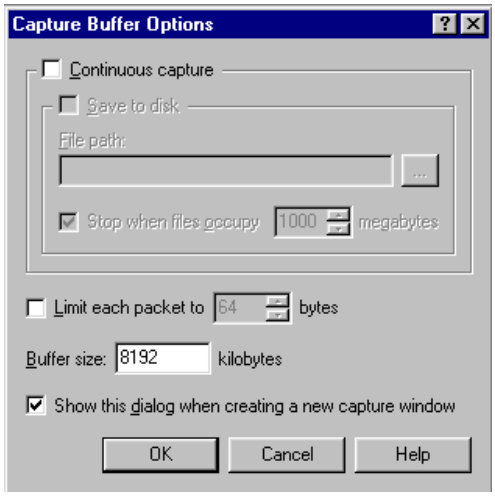


Figure 2-1: Select Capture > Start Capture from the menubar and enter 8192 to configure an 8 MB trace buffer for EtherPeek.

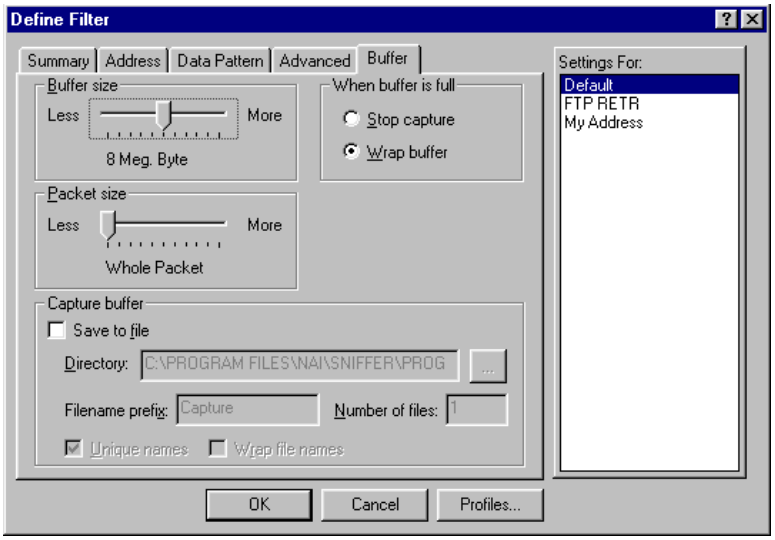


Figure 2-2: Select Capture > Define Filter and click on the Buffer tab to configure an 8 MB trace buffer for Sniffer.

You may need to stop the analyzer manually if it was set up in cyclical mode (where the analyzer continues to capture packets and cycles older packets out of the buffer).

Ok, so what do you do with 8 MB of packets? On the Sniffer Pro, I recommend that you begin to scroll through the summary screen to get a general feel for your communications. On the EtherPeek, I suggest you look at the statistics summary. As you scroll through the captured packets and statistics, answer the following questions:

- What is the most common protocol?
- What types of patterns appear?
- Does anything feel 'out of the ordinary?' (You'll get used to identifying strange traffic patterns as you practice more.)

Let's give it a try... check out the summary information shown in Figure 2-3.

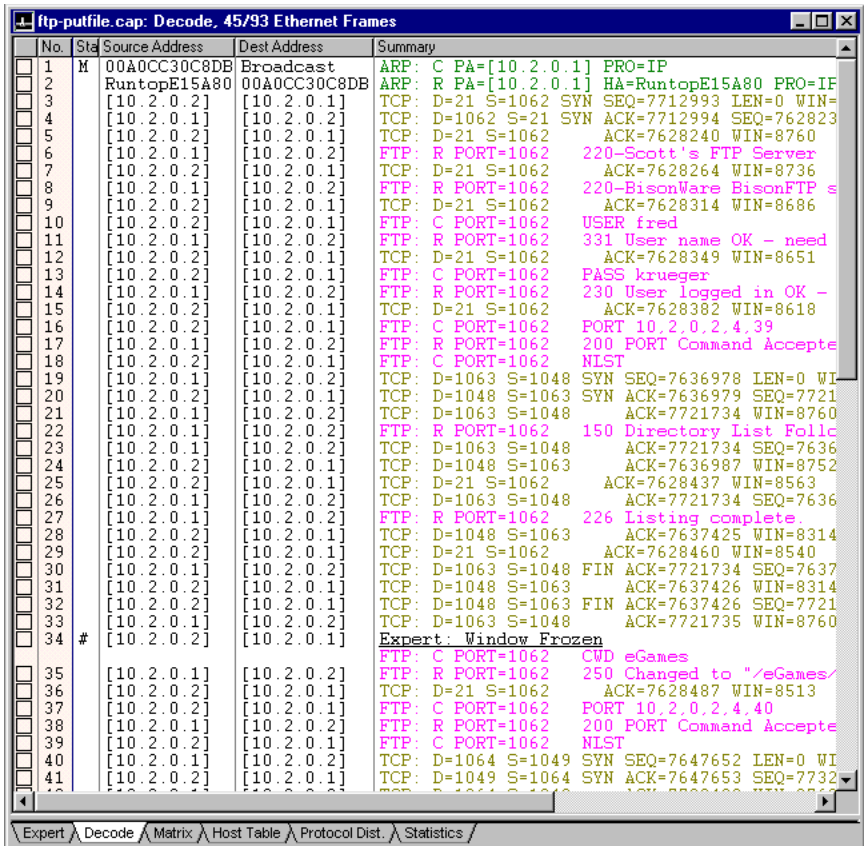


Figure 2-3: Scrolling through the Sniffer summery screen.

In this communication, you can see that there's an FTP communication going on and the user's name and password are cruising the network in plain text. We can see the Sniffer Expert system noted that the TCP window froze for a moment.

For more information on TCP windowing, refer to "TCP/IP Analysis and Troubleshooting" available through podbooks.com. -- Laura

If something really strange happened during the "laying on of hands," you might have noticed an alarm. For example, Figure 2-4 shows the Sniffer alarm log from one onsite session.

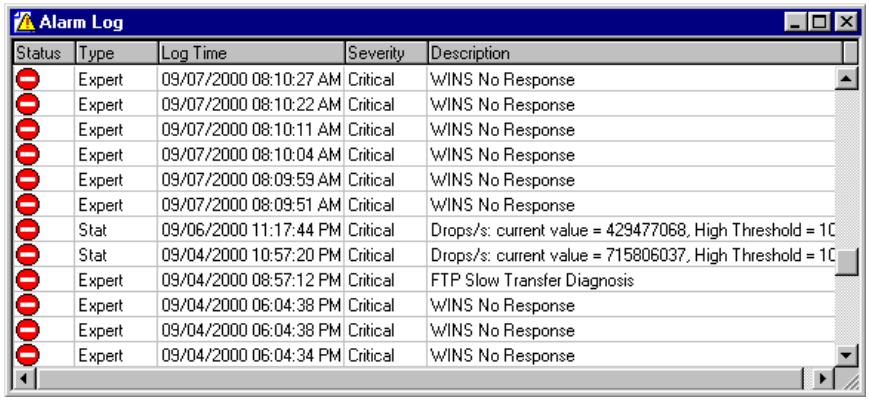

Alarm Log
Status
Type
Log Time
Severity
Description
Expert 09/07/2000 08:10:27 AM Critical WINS No Response
Expert 09/07/2000 08:10:22 AM Critical WINS No Response
Expert 09/07/2000 08:10:11 AM Critical WINS No Response
Expert 09/07/2000 08:10:04 AM Critical WINS No Response
Expert 09/07/2000 08:09:59 AM Critical WINS No Response
Expert 09/07/2000 08:09:51 AM Critical WINS No Response
Stat 09/06/2000 11:17:44 PM Critical Drops/s: current value = 429477068, High Threshold = 1C
Stat 09/04/2000 10:57:20 PM Critical Drops/s: current value = 715806037, High Threshold = 1C
Expert 09/04/2000 08:57:12 PM Critical FTP Slow Transfer Diagnosis
Expert 09/04/2000 06:04:38 PM Critical WINS No Response
Expert 09/04/2000 06:04:38 PM Critical WINS No Response
Expert 09/04/2000 06:04:34 PM Critical WINS No Response

Figure 2-4: The alarm log indicates that someone is looking for a WINS reply.

In the next step you'll begin to further classify the traffic and document the trends.

Classify Traffic and Document Trends

Now is the time to start focusing on specific areas in the trace buffer. The best way to do this is to view trend information and examine graphs and charts that correspond to the trace buffer.

What types of trends are most interesting? Here is a good checklist to follow:

- Protocol distribution
- Top 10 most active devices
- Packet size distribution
- Utilization trends
- Packets per second trends
- Error packets
- Broadcast and multicast traffic

Gather and print all of these trends for the traffic you've examined. Many times, they can help you identify strange traffic flows and faults.

This is when you really want an analyzer that has great graphical display capabilities. When Network Associates purchased NetXRay from Cinco Systems, they embraced the graphical display style that was part of the NetXRay product. EtherPeek always had a graphical format because it was based on the Macintosh originally.

Figure 2-5 shows some of the graphical capabilities of the Sniffer - excellent! You will use these graphs and charts to document your findings later in this chapter.

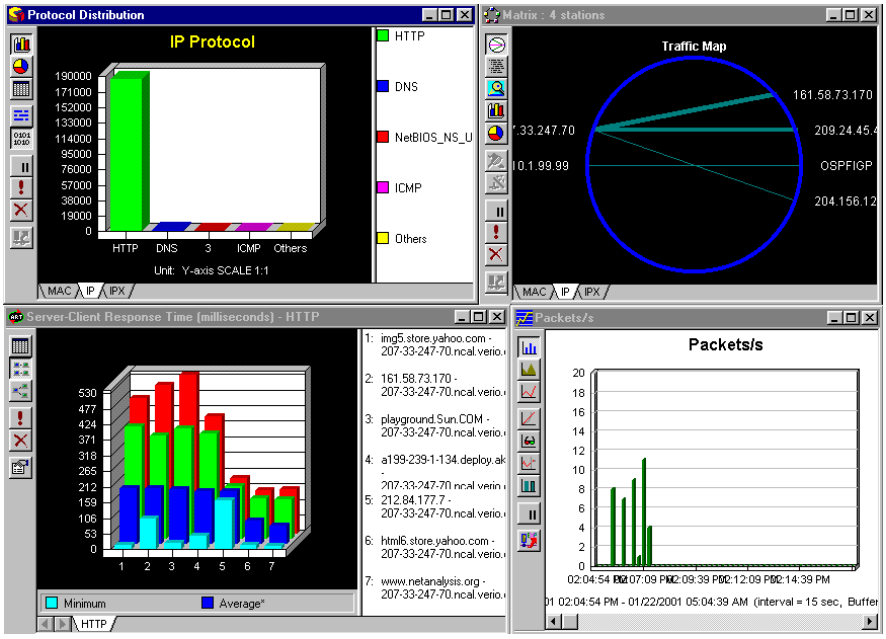


Figure 2-5 Sniffer provides numerous charts and graphs to help you classify the traffic trends.

What types of traffic trends did you see on your network?

Identify Possible 'Hot Spots'

Scroll through the summary screen that displays your network traffic. Did you notice anything that seemed strange? Really - it's ok to trust your gut feelings on this. Naturally, the analyzer alarm system can alert you to unusual conditions on the network, but your own intuition can also guide you to possible problems on the network.

- Too many broadcasts?
- Unanswered requests?
- Unrecognized packet structures?
- Unrecognized sources?

Focus in on Specific Conversations and Packet Types

When you suspect a network problem, consider focusing in on the traffic to and from the one device. Here are some examples of conversations that should be examined:

- *The network is experiencing excessive broadcasts*. Filter on all traffic to the broadcast address and find the most active device (the one that is broadcasting most often). Then classify the broadcast types to determine its purpose (RIP broadcasts to find a route, SAP broadcasts to find a service, NetBIOS broadcasts just to be bothersome). Typically broadcast storms are caused either by a device performing an unsuccessful lookup or by a device blasting information out to all devices on the network.
- *There is a high number of unnecessary traffic on the network*. For example, unanswered routing queries, excessive 'watchdog' or connection keep-alive sequences are just wasteful on the network.
- *A client appears to have numerous failure replies*. Filter on packets to and from that client device to create a nice packet-by-packet view of what the client has been doing. Perhaps the client mapped a resource to an incorrect location and it cannot find what it is looking for there.
- *An unusual amount of unknown/undecoded traffic is found on the network*. Consider filtering on some unique field value (perhaps the type field or the port field value). Now you can find the specific conversations that are using this undecoded communication. You might find some proprietary communication going on (for example, several router vendors perform a proprietary discovery process to locate other devices from that same vendor on the network).

Figure 2-6 shows one of those unusual conversations between two devices.

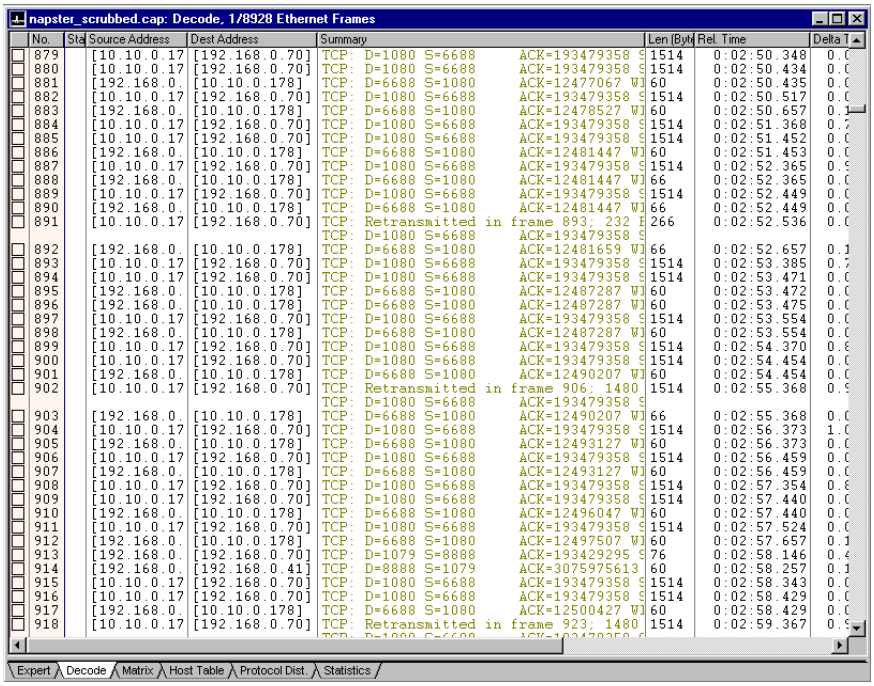


Figure 2-6: This file transfer appears to be sending only one packet at a time before requiring an acknowledgment.

This trace shows a strange communication sequence indeed... it appears that data is being transferred across the network over TCP, but only one packet at a time. Further, it appears there are some retransmissions occurring. These retransmissions indicate missing packets in the TCP communication.

For more dirt on TCP -specific communication issues, refer to "TCP/IP Analysis and Trouble-shooting" from podbooks.com. -- Laura

Certainly, this is a tremendously inefficient way to transfer data. Ugh. This is a Napster session, by the way (Napster v2.0 Beta 9.6). The Napster application is one of the ugliest on the network today -- you want to use up bandwidth and processing power, just let those folks have access to Napster!

The next paragraphs will describe a great trick that you should know how to do -- pattern analysis.

All There is to Know About Pattern Analysis

Pattern analysis is the process of identifying the pattern of requests and replies between two devices. For example, back in Figure 2-6, the pattern was:

- Request
- Reply

This is a normal pattern used for commands and requests for small amounts of data. Figure 2-7 shows the 'good' data patterns.

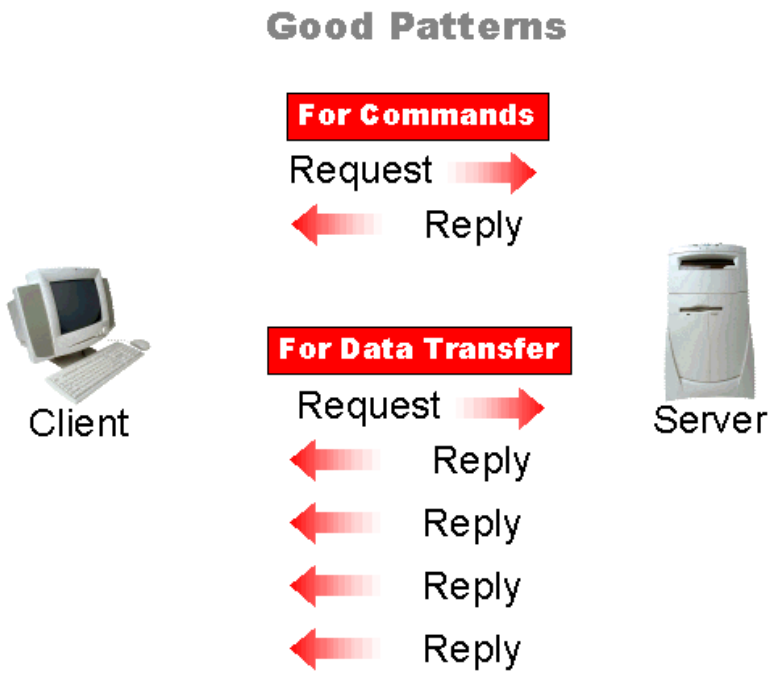


Figure 2-7: Good Patterns.

The first pattern depicts a typical command type of communication. The second pattern shows a type of pattern often seen for file transfers, such as TCP-based or IPX burst mode based file transfers.

Figure 2-8 shows the bad communication patterns. The top pattern shows a retry that is most likely caused by a timeout. The second pattern shows duplicate requests and replies that can be seen with frame type mismatches.

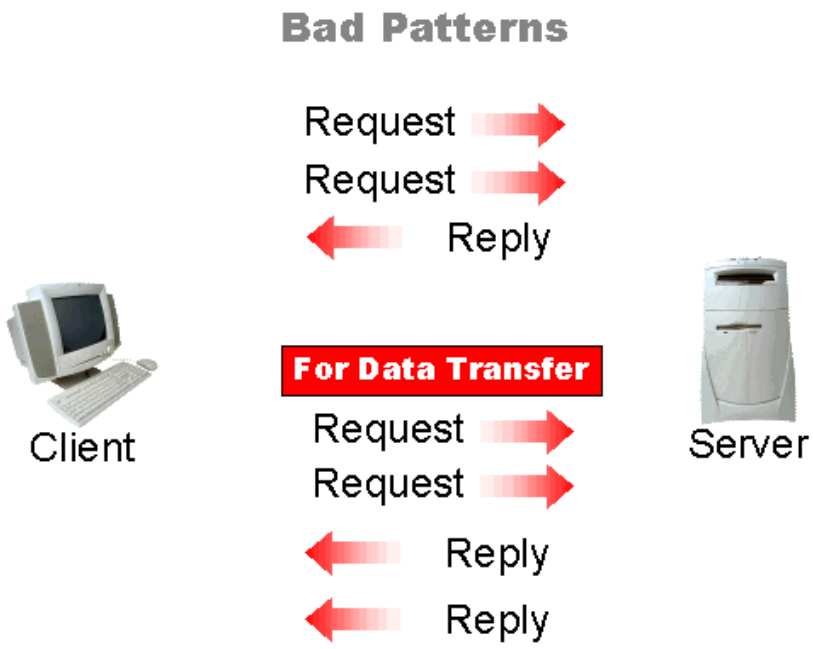


Figure 2-8: Multiple consecutive requests indicate a problem.

Pattern analysis is really quite simple. I recommend that you spend some time looking through the traces included with this book and the additional traces at www.packet-level.com and note the typical patterns seen in the communications.

Document Your Findings

It is important to document your findings in a graphical and textual format. One of the techniques I use is 'screenshot captioning' which makes for a very nice, graphical report. Of course, you can add as much depth as the reader requires -- you may even want to attach a disk with pertinent trace files. Figure 2-9 shows a sample of a screenshot captioned report page.

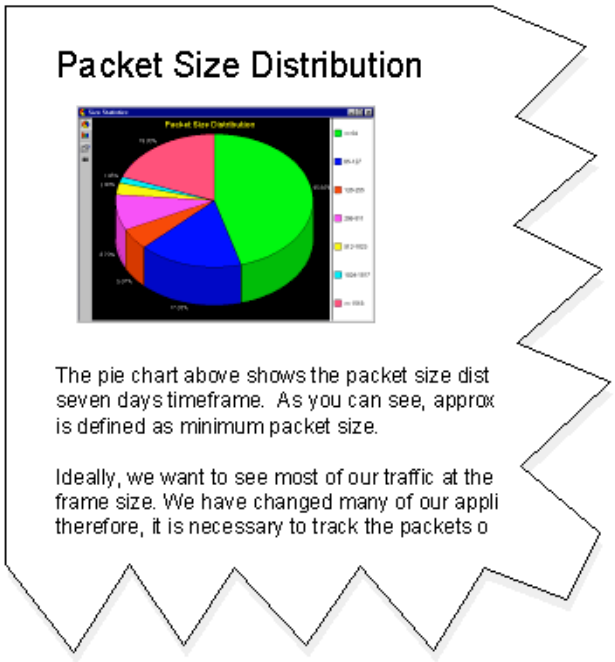

Packet Size Distribution
The pie chart above shows the packet size dist
seven days timeframe. As you can see, approx
is defined as minimum packet size.
Ideally, we want to see most of our traffic at the
frame size. We have changed many of our appli
therefore, it is necessary to track the packets o

Figure 2-9: Use graphical elements to make your report readable.

Not everyone will want to wade through numerous pretty pictures with no substance, so make sure you provide all your analysis findings in written format (don't assume the reader can interpret screenshots).

I always preface my reports with a 'summary' page that contains the content findings in bullet point format. This enables upper management or results-oriented people to quickly get the point of the report without reading the inside materials. - Laura

Now that we've covered the basic processes of analyzing your network from the 'laying on of hands' process through documentation, it is time to focus on more detailed areas of viewing captured packets.

Reading and Understanding the Summary Window

The analyzer summary window gives a brief interpretation of each packet in the trace buffer. In many instances, this is as far as you need to look to analyze network traffic patterns and identify possible problems. In this section, we'll look at how to use the data shown in each column of the summary window.

Remember that every analyzer is slightly different - refer to the analyzer documentation for more detail on areas not covered in this section. -Laura

Figures 2-10a and 2-10b show the summary window of the Sniffer split into two parts. The Sniffer summary screen includes the following columns:

- Marker Box
- Number
- Status
- Source Address
- Destination Address
- Summary
- Relative Time
- Length
- Delta Time
- Absolute Time

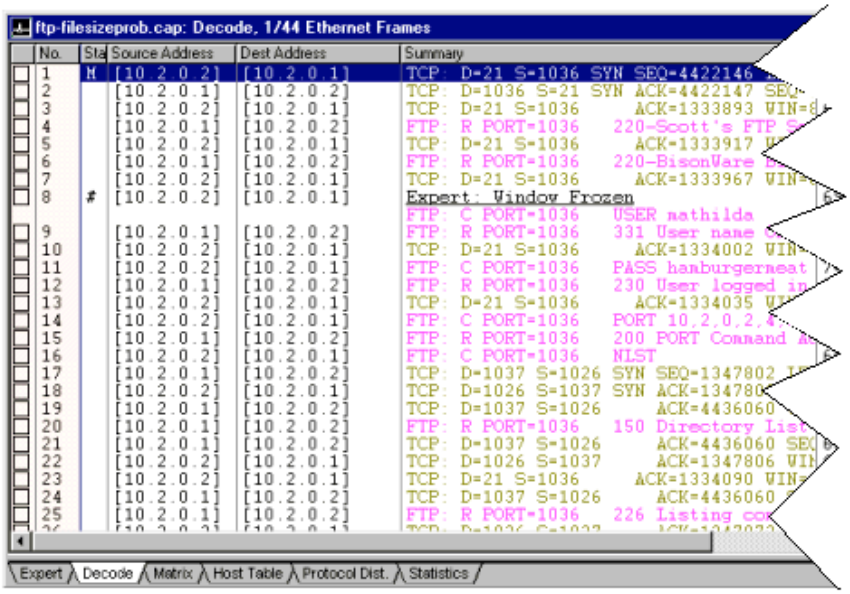


Figure 2-10a: The Sniffer summary screen.

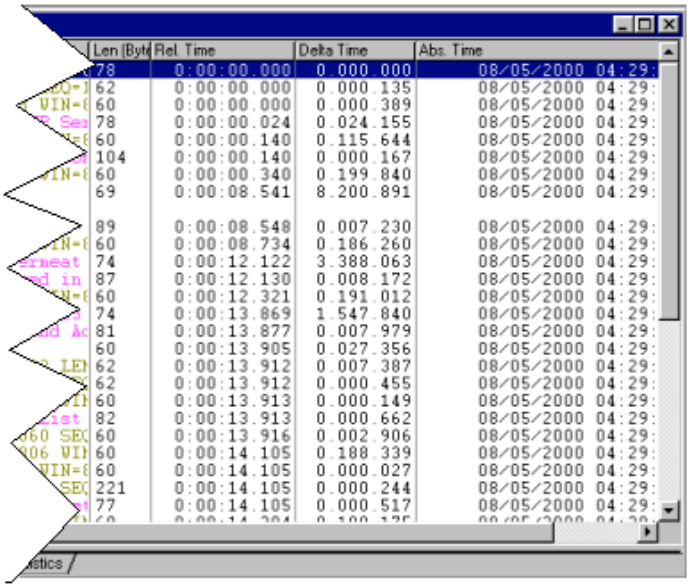


Figure 2-10b: The Sniffer summary screen (continued).

Figure 2-11 shows the EtherPeek summary window. The Sniffer summary screen includes the following columns:

- Packet Number
- Source Address
- Source Port

- Destination Address
- Destination Port
- Packet Size
- Protocol
- Plug-In Information

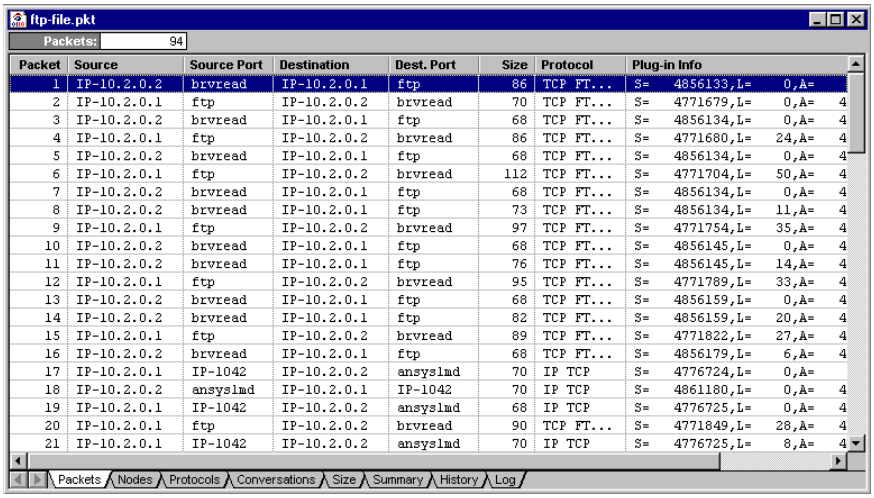

ftp-file.pkt
Packets: 94
Packet Source Source Port Destination Dest. Port Size Protocol Plug-in Info
1 IP-10.2.0.2 brvread IP-10.2.0.1 ftp 86 TCP FT... S= 4856133,L= 0,A=
2 IP-10.2.0.1 ftp IP-10.2.0.2 brvread 70 TCP FT... S= 4771679,L= 0,A= 4
3 IP-10.2.0.2 brvread IP-10.2.0.1 ftp 68 TCP FT... S= 4856134,L= 0,A= 4
4 IP-10.2.0.1 ftp IP-10.2.0.2 brvread 86 TCP FT... S= 4771680,L= 24,A= 4
5 IP-10.2.0.2 brvread IP-10.2.0.1 ftp 68 TCP FT... S= 4856134,L= 0,A= 4
6 IP-10.2.0.1 ftp IP-10.2.0.2 brvread 112 TCP FT... S= 4771704,L= 50,A= 4
7 IP-10.2.0.2 brvread IP-10.2.0.1 ftp 68 TCP FT... S= 4856134,L= 0,A= 4
8 IP-10.2.0.2 brvread IP-10.2.0.1 ftp 73 TCP FT... S= 4856134,L= 11,A= 4
9 IP-10.2.0.1 ftp IP-10.2.0.2 brvread 97 TCP FT... S= 4771754,L= 35,A= 4
10 IP-10.2.0.2 brvread IP-10.2.0.1 ftp 68 TCP FT... S= 4856145,L= 0,A= 4
11 IP-10.2.0.2 brvread IP-10.2.0.1 ftp 76 TCP FT... S= 4856145,L= 14,A= 4
12 IP-10.2.0.1 ftp IP-10.2.0.2 brvread 95 TCP FT... S= 4771789,L= 33,A= 4
13 IP-10.2.0.2 brvread IP-10.2.0.1 ftp 68 TCP FT... S= 4856159,L= 0,A= 4
14 IP-10.2.0.2 brvread IP-10.2.0.1 ftp 82 TCP FT... S= 4856159,L= 20,A= 4
15 IP-10.2.0.1 ftp IP-10.2.0.2 brvread 89 TCP FT... S= 4771822,L= 27,A= 4
16 IP-10.2.0.2 brvread IP-10.2.0.1 ftp 68 TCP FT... S= 4856179,L= 6,A= 4
17 IP-10.2.0.1 IP-1042 IP-10.2.0.2 ansyslmd 70 IP TCP S= 4776724,L= 0,A=
18 IP-10.2.0.2 ansyslmd IP-10.2.0.1 IP-1042 70 IP TCP S= 4861180,L= 0,A= 4
19 IP-10.2.0.1 IP-1042 IP-10.2.0.2 ansyslmd 68 IP TCP S= 4776725,L= 0,A= 4
20 IP-10.2.0.1 ftp IP-10.2.0.2 brvread 90 TCP FT... S= 4771849,L= 28,A= 4
21 IP-10.2.0.1 IP-1042 IP-10.2.0.2 ansyslmd 70 IP TCP S= 4776725,L= 8,A= 4
Packets Nodes Protocols Conversations Size Summary History Log

Figure 2-11: The EtherPeek summary screen.

In the next section, we'll take a look at what each of these columns can tell you about your network. We'll focus on a combination of all the columns offered by Sniffer and EtherPeek.

Marker Box Column/Marking Capability

Both EtherPeek and Sniffer offer ways to select a subset of the trace buffer. This effectively enables you to apply a quick filter on the packets and then save the subset.

Figure 2-12 gives you an example of how to use this feature on the Sniffer. Notice that I've checked off certain packet's marker boxes on the summary screen. Next, I right mouse clicked to get the menu box and clicked on 'save selected.' This enables me to get a separate window with just the packets I've marked.

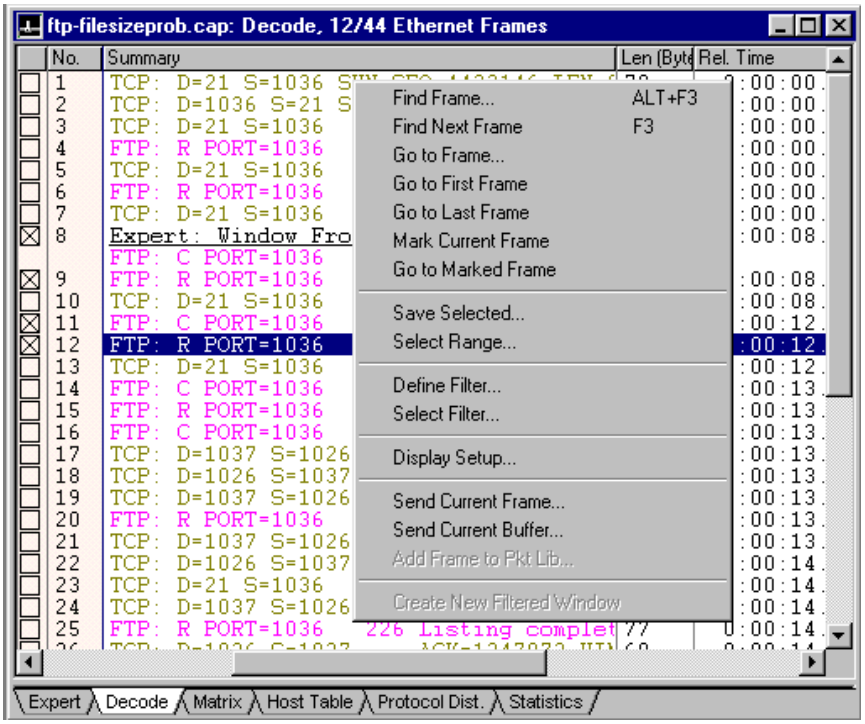


Figure 2-12: Right mouse click to get the Display menu. [Sniffer]

You don't have to pick sequential packets - you can pick the first and last packet, for example. Then you can save just those packets. For example, perhaps you caught your password crossing the cabling system -- you could click on and save the packets that disclose your password to show them to management as proof that your network is not secure. Of course, then you really need to change your password (grin).

It should be noted that you can easily select a series of packets with EtherPeek, as well, even though there's no Marker column. On EtherPeek, you simply click anywhere on the summary line to select the packet. Again, a right mouse click displays a menu allowing you to save those selected packets.

Number Column

All analyzers have a number column in their summary windows. This is simply the number of packet in the order it was received in the trace buffer. Packets in the trace buffer cannot be reordered by the type of packets, packet size or any other characteristic. They remain in the 'receive order.'

Status Column

The Status column on the Sniffer indicates if a packet is associated with a symptom or diagnosis or if the packet is considered an error packet. It can also denote a packet that is time marked (all column times are based on this packet's receipt time).

The following list defines the possible Status field values:

#	symptom/diagnosis associated with packet
M	time marked packet
Trigger	packet triggered an event
CRC	CRC error packet (see Chapter 4)
Jabber	Jabber error packet (see Chapter 4)
Runt	Runt or undersized packet (see Chapter 4)
Fragment	Fragment, not a true packet (see Chapter 4)
Oversize	Oversized packet (see Chapter 4)
Collision	Packet damaged by a collision (see Chapter 4)
Alignment	Packet alignment is invalid (see Chapter 4)
Duplicate Address	Token Ring address conflict (see Chapter 4)
Frame Copied	Token Ring packet not copied (see Chapter 4)

Source Address Column

This column identifies the device that sent the packet. This field may contain the following addresses:

Hardware address: If the packet does not contain a network-layer address (ARP, for example), the hardware address -- also referred to as the Media Access Control address or Network Interface Card address -- is shown in this field.

Network address: If a network address exists in the packet, the analyzers typically show this address. For example, in an HTTP (Hypertext Transport Protocol) packet used to web browse, the IP (Internet Protocol) header contains the IP address of the source. This column will, therefore, contain that network-layer address.

Name: Decent analyzers can either discover device names (such as NetBIOS names or server names) or allow you to manually enter names to help identify devices.

Consider the following source address formats that can be displayed by the Sniffer:

72132311.0000A0C92322	IPX network/node address combination
10.1.2.3	IP address
0800092341394562_PS1	Print server SAP identifier
FS1-C983	File server name

For more information on the network and MAC-layer source addresses, refer to Appendix B, "The Basic Flow of Data."

You should be very comfortable with the address process and the fact that the MAC layer is stripped off and replied by routers or routing processes. If not, I highly recommend you read Appendix B. -Laura

Source Port

The source port field either indicates the process that sent the packet or a temporary dynamic port number to be used for the duration of the communication process.

Destination Address Column

The destination address column is the same as the source address column, except that it defines the address that the packets are supposed to go to. It can include broadcast and multicast addresses or unicast (single device) addresses.

One of the common display filters you'll apply is one that filters on packets sent to the broadcast or multicast addresses. You should be familiar with the following possible destination addresses:

MAC Layer Broadcasts

Broadcast	0xFF-FF-FF-FF-FF-FF
Token Ring Broadcast	0xC0-00-FF-FF-FF-FF

IPX Broadcast/Multicast Addresses

IPX Broadcast	0x00-00-00-00:FF-FF-FF-FF-FF-FF
NLSP Multicast IEEE 802.3	0x09-00-1B-FF-FF-FF
NLSP Multicast IEEE 802.5	0xC0-00-10-00-00-00
NLSP Multicast FDDI	0x09-00-1B-FF-FF-FF

Internet Broadcast/Multicast Addresses (Decimal)

All Nets Broadcast	255.255.255.255
OSPF All Routers	224.0.0.5
OSPF Designated Routers	224.0.0.6
RIP2 Routers	224.0.0.9
IGRP Routers	224.0.0.10
Service Location Multicast	224.0.1.22
SLP Directory Agent	224.0.1.35
Cisco Discovery	224.0.1.40

For a complete list of assigned multicast addresses, refer to www.iana.org*.-- Laura*

Destination and Source Ports

The port fields indicate the process or application that the packet is sent to. To obtain the full list of assigned port numbers for TCP/IP communications, again refer to www.iana.org.

Summary Column

Scroll through a trace and review this column to find out the purpose of the packets shown.

Analyzers use different terminology in the summary section which gives a general idea of what's going on in the packet sequence.

Remember, the summary column gives you an overview of the packet only - if you want more detail, you'll have to go to the decode level (defined in the next section).

Length Column

The length column indicates the length of the packet in bytes. On most analyzers, this field does not include the 4-byte CRC found at the end of the Ethernet packet structures.

I always do a quick check down this column to see what the common frame size is. You can also use the packet size distribution graphs to see a nice representation of your packet sizes.

Larger packets are more efficient on the network because you have more payload (data) for each header built and processed. You can see if there is a 'ceiling' on packet size as you browse through the summary window's length column.

Time Columns

There are three common time columns used in analyzer summaries:

- Relative Time
- Delta (or Interpacket) Time
- Absolute Time

The Relative Time indicates how much time has passed between the arrival of this packet and the first packet (or a time-marked packet) in the trace buffer. The format is usually defined as:

HH:MM:SS.ss

Where HH = hour

MM = minute

SS = seconds

ss = milliseconds (hundredths of a second)

This field is really quite useful, especially on the Sniffer where you can time mark a packet. By time marking a packet in the trace buffer and finding a later packet, you can easily determine the difference in time between the two packets. For example, perhaps you captured a trace that contains a login sequence. You can time mark the first packet of the login sequence and scroll down to the last packet generated by the sequence to figure out how long the process takes.

Delta (Interpacket) Time Column

This column indicates the time between one packet in the trace buffer and the previous packet in the trace buffer. You can use this column to see how fast the request/reply pattern is executed.

Try it out! Look at the delta between a request and a reply when you ping a local device. Now ping a device that is far away. What is the delta time between your request and the reply? This delta timer can help you estimate the 'time cost' of putting an application farther away from users. Simply multiply the average delta time by the number of request/reply sets to get an idea of how long it will take to access some remote process.

For example, consider an application launch that requires 350 request/reply sets and your average delta time between a request and a reply is .002 seconds. Multiply the number of sets by the average delta to get an approximate time to load the application across the network. What if you moved the application to the other side of a WAN link that typically indicates a delta time of .033 seconds?

Consider Figure 2-13, where we did some very simple math to find out that it will take approximately 11.9 seconds to launch the application across the WAN link! Totally unacceptable!

Not all application launch sequences use this single request/single reply pattern, however. In many cases you will see nice bursts of data (multiple replies to a single request). This formula is only used for a very basic guideline. You can measure the exact timestamp by testing the application and viewing the relative time between the first and last packet of the launch sequence.

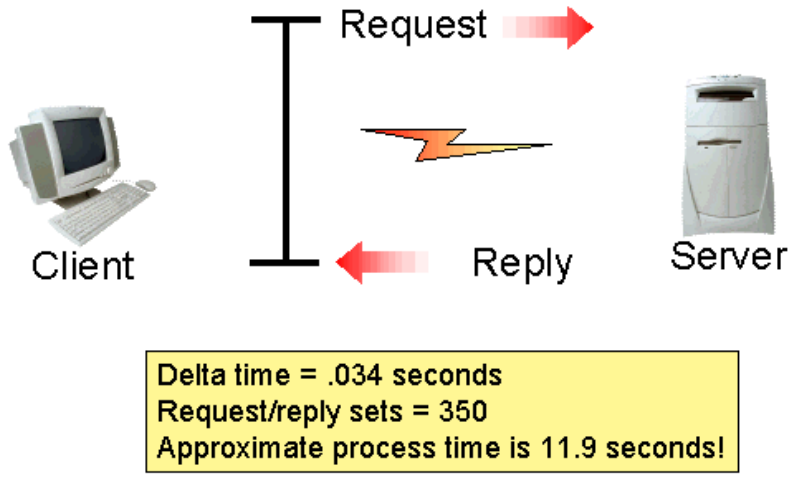


Figure 2-13: You can figure out a process timestamp once you have the average delta and the number of request/reply sets.

Absolute Time Column

The absolute timestamp column indicates the actual time of day that the packet arrived in the trace buffer. The time is based on the local system clock.

It is always good to have a stationary reference for time as you perform your analysis. Many times I have found myself going through a trace file after an experiment only to find that the timestamp indicated the time was 3:00 in the morning! -Laura

---

If you are new to analysis, you might find the delta and relative timestamping a bit confusing - in that case, you can always use the absolute timestamp.

Analysis Tasks You Can Do Today

It's time to take a test drive on your network. This section gives you some very basic guidelines for performing various analysis tests.

One-Hour Baseline (Quick Look)

Plug your analyzer into the network (make certain to span ports if you're connecting into a switch). Start capturing all packets with a cyclical buffer. Let it run for one hour.

After the hour, gather the following data:

1. Utilization trend (screen capture graph)
2. Packets trend (screen capture graph)
3. Most active stations (the Host Table tab on the Sniffer or the Conversations tab on the EtherPeek)
4. Error rate (screen capture graph)
5. Broadcast Rate (as a percentage of total traffic)

CAUTION *Ok... time to talk about security. If you are going to leave an analyzer unattended, ensure it is behind a locked door. It's your data cruising along there and not all of it is secured through encryption methods. Analyzers that can transmit (such as Sniffer and EtherPeek) are especially dangerous because anyone can easily perform a Denial of Service attack with a transmitting analyzer. See Chapter 4 for more information on these attacks. -Laura*

Categorize Your Broadcast Traffic

Set up a capture filter to look for packets addressed to the broadcast address (0xFF-FF-FF-FF-FF-FF). Once you've captured a full 8 MB of data look at the trace buffer to see what types of broadcast traffic is on your network.

By applying several display filters, you can determine how many packets are:

- ARP broadcasts (filter on Ethernet type 0x0806)
- DHCP broadcasts (filter on DHCP destination port 0x67)
- NetWare SAP broadcasts (filter on the SAP destination socket 0x0452)
- NetWare RIP broadcasts (filter on the RIP destination socket 0x0453)

- NetWare NLSP broadcasts (filter on NLSP destination socket 0x9001)
- Other broadcasts (subtract the above numbers from the total number of packets captured)

How many broadcasts are too many? Capture all traffic (no filter) during an active time and filter out broadcasts. If you find that the filtered packet number is 10% of the total number of packets captured, you have a high broadcast rate. Remember, however, that every network is different. You should be most concerned about broadcasts that occur during peak production times.

Identify Errors

Set up your analyzer to run through the day. At the end of the day, examine the error log to see what types of errors occur on your network. Are there any alarms that were triggered more than once per minute?

Document your network error rate. Refer to Chapter 4, "Identifying Typical Problems" for more details on Ethernet and Token Ring errors.

TCP/IP: Check out Your ICMP Traffic

Set up a capture filter for all ICMP traffic. Define the type of ICMP traffic on your network.

Go to www.ietf.org and download RFC 792, "Internet Control Message Protocol." Read up on what the ICMP traffic is telling you. The ICMP type list is included in Chapter 4.

IPX/SPX: Check out IPX Burst Mode Communications

Set up a capture filter to look for traffic between a client and server that is using IPX (perhaps your own desktop system).

Copy a large file down from your server to your client.

Stop the trace and look at the transfer process. Did you see burst mode communications (request, reply, reply, reply)?

You can set up a display filter on all packets to the NCP type field value 0x7777 to view all burst mode packets.

Capture Your Bootup Sequence

To capture a bootup sequence, you should set up a test station and filter on the test station's address. Turn on your test system and DON'T TOUCH THE KEYBOARD.

This is a major problem for most troubleshooters - we all want to touch the keyboard and 'be one' with our computers. Try to resist so you can get a nice natural bootup sequence. -Laura

Wait until the packet rate has dropped down to zero packets/second steadily and no further processing is going on at the desktop. Then open the trace buffer and check out your packet-by-packet bootup sequence.

- Anything strange there?
- How long did it take?
- How many packets did it take?
- How many broadcasts did it generate?

Document your bootup sequence.

Capture Your Login Sequence

Start your analyzer with a capture filter on your desktop MAC address again. Login to the network and then remove your hands from the keyboard. DON'T TOUCH THE KEYBOARD AGAIN.

You may need to input your password - which is ok. Just don't launch an application or anything like that. -Laura

Wait until the packet rate has dropped down to zero packets/second steadily and no further processing is going on at the desktop. Then open the trace buffer and check out your packet-by-packet login sequence.

- Anything strange there?
- How long did it take?
- How many packets did it take?
- How many broadcasts did it generate?
- How many failures?
- Is your password encrypted?

I hope you take the time to try these exercises out.

One of the great benefits of analyzing your own traffic is that you can learn exactly how things work with a graphical step-by-step display of the process.

No matter how many books you read, you can't replace the experience you gain from seeing it with your own eyes.

Chapter 2 Test

Here we go. One portion of this test requires you to download the latest EtherPeek demo from www.wildackets.com (or use your fully-registered EtherPeek or Sniffer) to answer the following questions.

2.1 What is the difference between the absolute and relative timestamps?

_____

_____

2.2 You have caught an unrecognized multicast packet in your trace buffer. Where should you go to lookup the registration for this address?

_____

2.3 Your network uses full-duplex hubs and switches. How can you connect your analyzer to ensure that you capture all traffic that flows through the TX and RX channels?

_____

2.4 You want to capture all the traffic to and from your key router as you download music using Napster (or a similar music-sharing program). How does this task affect the buffer size?

_____

2.5 Analysis Test

EtherPeek users: Download *ftp-file.pkt* from www.packet-level.com/tracefiles.

Sniffer users: Download *ftp-file.cap* from www.packet-level.com/tracefiles.

Open the file and answer the following questions:

What format is the source address in (MAC-address, network address or name)?

_____

What application is used in Packet 1?

_____

What is the length of this packet?

_____

There are lots of trace files and references online at www.packet-level.com. Download the SMTP, telnet and FTP trace files and examine their summary information, as well. Very interesting stuff.

CHAPTER 3

Packet Structures and Decodes

This chapter gets you up to speed on packet layouts and general decoding. If you still feel a bit wobbly on the whole communications stuff, check out Appendix B, "The Basic Flow of Data."

Many people shy away from analysis because of the complex appearance of the packet decodes and detail screens. Once you complete this section, you will feel at ease with the packets, headers and the resource information that provides interpretations.

Those Decodes

The best way to deal with packet decodes is to jump right in. Most analyzers show you a bit of the packet decode as one of the trace windows (directly below the summary window). They often show you the hexadecimal version of the packet as well (thereby satisfying your intense curiosity of the hex format of your data, eh?).

Figure 4-1 shows the EtherPeek full decode and hex windows. When you highlight a field in the decode window, the corresponding hex digits are also highlighted, as shown.

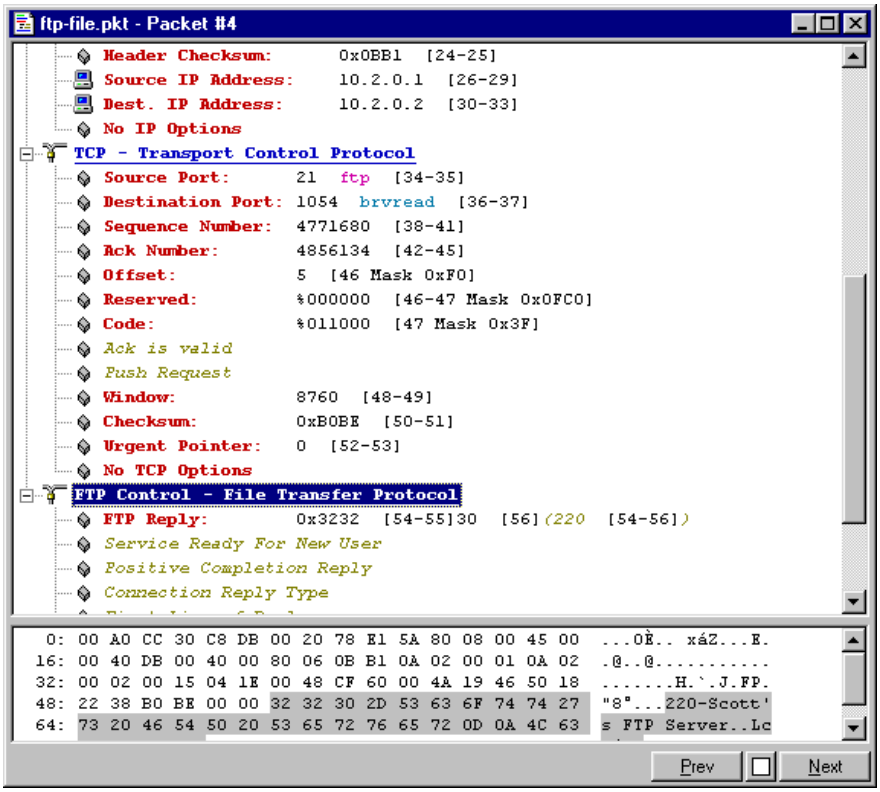

ftp-file.pkt - Packet #4
Header Checksum: 0x0BB1 [24-25]
Source IP Address: 10.2.0.1 [26-29]
Dest. IP Address: 10.2.0.2 [30-33]
No IP Options
TCP - Transport Control Protocol
Source Port: 21 ftp [34-35]
Destination Port: 1054 brvread [36-37]
Sequence Number: 4771680 [38-41]
Ack Number: 4856134 [42-45]
Offset: 5 [46 Mask 0xF0]
Reserved: %000000 [46-47 Mask 0x0FC0]
Code: %011000 [47 Mask 0x3F]
Ack is valid
Push Request
Window: 8760 [48-49]
Checksum: 0xB0BE [50-51]
Urgent Pointer: 0 [52-53]
No TCP Options
FTP Control - File Transfer Protocol
FTP Reply: 0x3232 [54-55]30 [56](220 [54-56])
Service Ready For New User
Positive Completion Reply
Connection Reply Type
0: 00 A0 CC 30 C8 DB 00 20 78 E1 5A 80 08 00 45 00 ...0È.. xáZ...E.
16: 00 40 DB 00 40 00 80 06 0B B1 0A 02 00 01 0A 02 .@..@.....
32: 00 02 00 15 04 1E 00 48 CF 60 00 4A 19 46 50 18 .....H.`.J.FP.
48: 22 38 B0 BE 00 00 32 32 30 2D 53 63 6F 74 74 27 "8"...220-Scott'
64: 73 20 46 54 50 20 53 65 72 76 65 72 0D 0A 4C 63 s FTP Server..Lc
Prev
Next

Figure 4-1: An FTP packet decoded by EtherPeek.

Decodes are built by identifying a protocol used in a packet and referencing the specifications to determine the field offsets and values/interpretations.

How Packets are Built

You communicate on the network by sending a series of packets that contain requests, replies, general information, connection management, etc. In this section, we'll look at the contents of a packet in the order in which it is built.

In this section, I have set EtherPeek to show the data offsets so you can see where data resides inside a packet. This capability is a great advantage when you are learning packet offsets and field lengths. To set up EtherPeek to display these offsets, click ***View*** *>* ***Packet Decoder*** *>* ***Show Data Offsets****. -- Laura*

The Application Defines the Data

The application you are working with (Navigator, Word, WS_FTP Pro, or whatever) builds the actual data that needs to fit in the packet. This data is often referred to as the 'payload.' In Figure 4-2, the 'data' is an FTP response code 220 with the FTP server welcome message ("Scott's FTP Server").

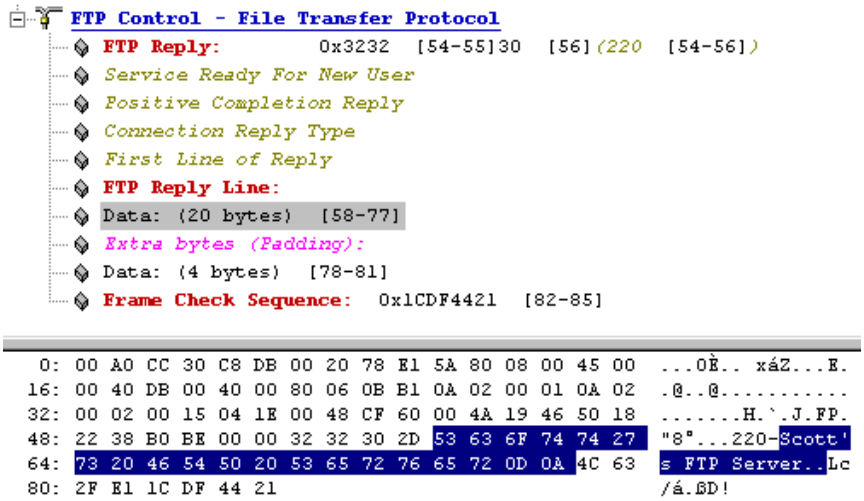

FTP Control - File Transfer Protocol
FTP Reply: 0x3232 [54-55]30 [56](220 [54-56])
Service Ready For New User
Positive Completion Reply
Connection Reply Type
First Line of Reply
FTP Reply Line:
Data: (20 bytes) [58-77]
Extra bytes (Padding):
Data: (4 bytes) [78-81]
Frame Check Sequence: 0x1CDF4421 [82-85]
0: 00 A0 CC 30 C8 DB 00 20 78 E1 5A 80 08 00 45 00 ...0È.. xáZ...E.
16: 00 40 DB 00 40 00 80 06 0B B1 0A 02 00 01 0A 02 .@..@.....
32: 00 02 00 15 04 1E 00 48 CF 60 00 4A 19 46 50 18 .....H.`.J.FP.
48: 22 38 B0 BE 00 00 32 32 30 2D 53 63 6F 74 74 27 "8°...220-Scott'
64: 73 20 46 54 50 20 53 65 72 76 65 72 0D 0A 4C 63 s FTP Server..Lc
80: 2F E1 1C DF 44 21 /á.ßD!

Figure 4-2: The application payload contains an FTP response.

As you can see in Figure 4-2, I have used EtherPeek to capture an FTP response.

For more details on FTP analysis and operations, refer to the "Advanced Network Analysis Techniques" podbooks and www.packet-level.com *where I placed a short article on FTP analysis (check the archives). -- Laura*

As we look more closely at Figure 4-2, you notice that I have highlighted the FTP heading in the full decode window. The corresponding bytes have been highlighted in the hex window. The hex window also provides the ASCII interpretation of the hex value, making it easy to determine what this FTP server is saying. The total amount of payload being sent 24 bytes -- 20 bytes of data plus four bytes of FTP command information. The command information is 220- (in hex, that's 0x3232-302D).

The application (FTP in this case), sends this data to the transport layer, TCP. If the application had been written to use connectionless services, it would send the data to UDP.

The Protocol Stack Inserts the Transport Layer Header

At this point, the transport layer header, TCP is placed in front of the FTP data. This TCP header plus data is called a *TCP segment*. You'll notice that we change the name of this bundle o' bits as we move it through the layers to get it out on the wire.

The source port number, 21, indicates that this packet is coming from the FTP process. The destination port number is a dynamic or temporary port number.

For the complete list of port numbers, refer to www.iana.org. *-- Laura*

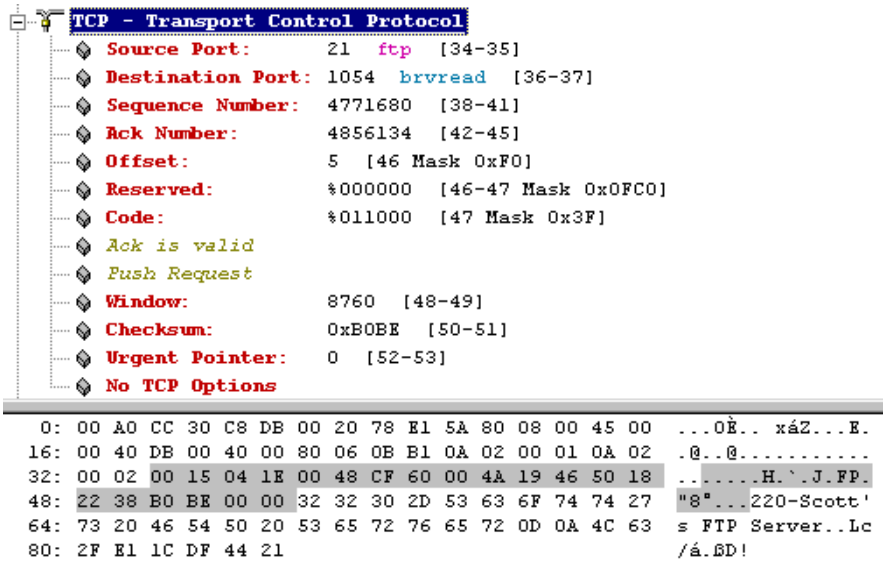

TCP - Transport Control Protocol
Source Port: 21 ftp [34-35]
Destination Port: 1054 brvread [36-37]
Sequence Number: 4771680 [38-41]
Ack Number: 4856134 [42-45]
Offset: 5 [46 Mask 0xF0]
Reserved: %000000 [46-47 Mask 0x0FC0]
Code: %011000 [47 Mask 0x3F]
Ack is valid
Push Request
Window: 8760 [48-49]
Checksum: 0xB0BE [50-51]
Urgent Pointer: 0 [52-53]
No TCP Options
0: 00 A0 CC 30 C8 DB 00 20 78 E1 5A 80 08 00 45 00 ...0È.. xáZ...E.
16: 00 40 DB 00 40 00 80 06 0B B1 0A 02 00 01 0A 02 .@..@.....
32: 00 02 00 15 04 1E 00 48 CF 60 00 4A 19 46 50 18 .....H.`.J.FP.
48: 22 38 B0 BE 00 00 32 32 30 2D 53 63 6F 74 74 27 "8°...220-Scott'
64: 73 20 46 54 50 20 53 65 72 76 65 72 0D 0A 4C 63 s FTP Server..Lc
80: 2F E1 1C DF 44 21 /á.ßD!

Figure 4-3: The Source port 21 indicates that this is coming from the FTP process.

In Figure 4-3, I have highlighted the EtherPeek TCP heading -- this, in turn, highlights the entire 20-byte TCP header in the hex window. The Offset field contains the value 5. This value is basically the TCP header length field (in 4-byte increments). Multiply this field by 4 to determine that the TCP header is 20 bytes long.

"Now, how the hell would I know that?" you're saying to yourself. Well... you need to refer to the specification, RFC 793. I've included the general TCP header structure and field definitions later in this chapter.

In this case, the EtherPeek folks could have interpreted this for you by inserting the words "TCP header is 20 bytes long" in the decode window, but they didn't.

In most cases, the decodes are complete enough that you know what the fields are used for (unless it's considered obvious, as in the case of the checksum fields).

Consider the 'window' field -- what's that all about? Refer to RFC 793 to obtain an interpretation of this field.

The Protocol Stack Inserts the Network Layer Header

At this point, the TCP segment is handed down to the IP protocol. The IP header is placed in front of the TCP header. The IP header contains a protocol field that indicates this packet came from TCP (the value 6 is assigned to TCP).

The IP header also contains the tremendously important source and destination IP addresses. These addresses are used by routers to forward packets through an internetwork.

Remember, if you are looking for the possible field values, go to www.iana.org. *If you are interested in the purpose of a field, go to the RFC at* www.ietf.org. *-- Laura*

In Figure 4-4, I have highlighted the IP header.

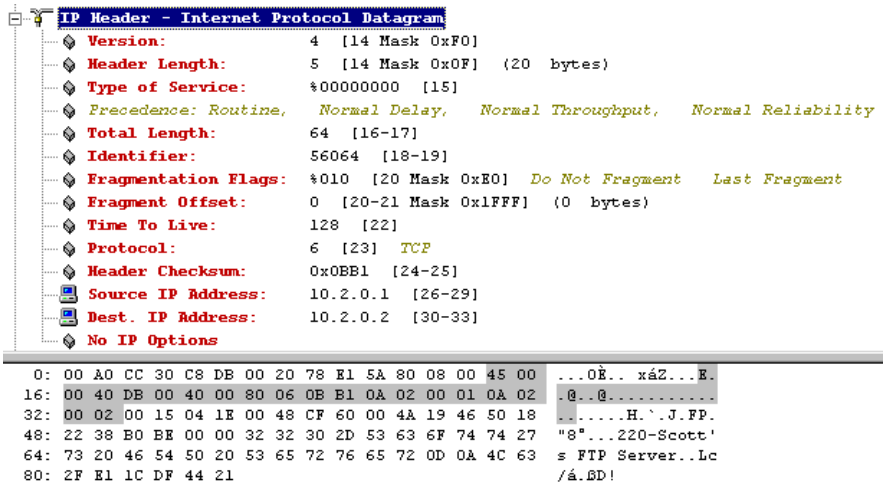

IP Header - Internet Protocol Datagram
Version: 4 [14 Mask 0xF0]
Header Length: 5 [14 Mask 0x0F] (20 bytes)
Type of Service: %00000000 [15]
Precedence: Routine, Normal Delay, Normal Throughput, Normal Reliability
Total Length: 64 [16-17]
Identifier: 56064 [18-19]
Fragmentation Flags: %010 [20 Mask 0xE0] Do Not Fragment Last Fragment
Fragment Offset: 0 [20-21 Mask 0x1FFF] (0 bytes)
Time To Live: 128 [22]
Protocol: 6 [23] TCP
Header Checksum: 0x0BB1 [24-25]
Source IP Address: 10.2.0.1 [26-29]
Dest. IP Address: 10.2.0.2 [30-33]
No IP Options
0: 00 A0 CC 30 C8 DB 00 20 78 E1 5A 80 08 00 45 00 ...0È.. xáZ...E.
16: 00 40 DB 00 40 00 80 06 0B B1 0A 02 00 01 0A 02 .@..@...........
32: 00 02 00 15 04 1E 00 48 CF 60 00 4A 19 46 50 18 ....H.`.J.FP.
48: 22 38 B0 BE 00 00 32 32 30 2D 53 63 6F 74 74 27 "8°...220-Scott'
64: 73 20 46 54 50 20 53 65 72 76 65 72 0D 0A 4C 63 s FTP Server..Lc
80: 2F E1 1C DF 44 21 /á.ßD!

Figure 4-4: The decoded IP header.

The IP Header Length field indicates that this IP header is 20 bytes long (again, the value in this field should be multiplied by 4 -- in this case, the EtherPeek decodes have done the math for you).

The IP header also contains a Total Length field that indicates the length of the packet from the beginning of the IP header to the end of the valid data (this does not include any padding -- we'll talk about padding pretty soon).

We can figure this out without this Total Length field:

IP header:	20 bytes
TCP header:	20 bytes
FTP command/data:	24 bytes
Total:	**64 bytes**

Aaaahhhh.... you can never have enough length fields <grin>. -- Laura

At this point, we have an *IP datagram*. This includes the IP header, TCP header and the data.

It's time to send this IP datagram to the LAN driver so the MAC (Media Access Control) header can be placed on the front.

The LAN Driver Inserts the MAC Header

This MAC header supplies the local addresses to get the packet from one device to another on the local network. The MAC header is

placed on the packet by the LAN driver after it performs some structure checks on the contents to ensure it is within an acceptable size range.

For example, on Ethernet networks, the minimum packet size is 64 bytes. This is based on a minimum data portion (after the MAC header) of 46 bytes. If the stuff coming to the MAC header is less than 46 bytes long, the MAC header is going to pad the data.

There are several different MAC header formats defined -- Ethernet II, Ethernet 802.3, Ethernet SNAP, Token Ring, Token Ring SNAP, FDDI, etc. Many of the MAC frames are defined by the IEEE (Institute for Electrical and Electronic Engineers). Check out www.ieee.org for more information on these frame types.

Figure 4-5 shows the decode of the Ethernet II frame that precedes the IP datagram.

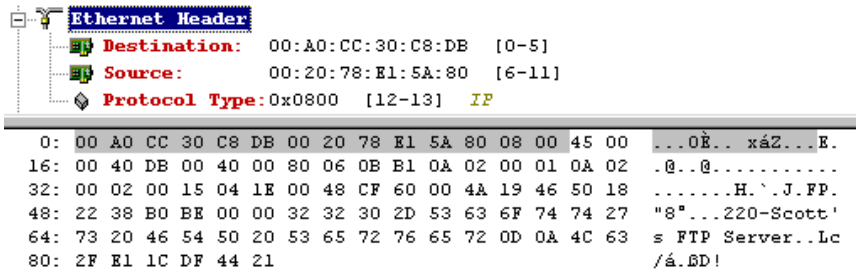


Figure 4-5: The Ethernet II header.

As you can see in Figure 4-5, the Ethernet II header only contains three fields:

- Destination Address
- Source Address
- Protocol Type

The Protocol Type field is used to indicate what protocol follows the MAC header. In this case, 0x0800 indicates that an IP header is following the MAC header. All decent MAC headers contain some flavor of a Type field.

The most commonly seen Type field values are:

0x0800	IP
0x0806	ARP
0x8137	IPX/SPX

Notice that I said 'decent' MAC headers... guess which header structure this leaves out? Remember the old Novell Ethernet_802.3 frame structure -- also referred to as Ethernet 'Raw?' That was an ugly one... it didn't have an protocol ID field -- it was built in the days when we - I'll accept some blame here - believed that IPX was the only protocol we'd see on a NetWare network. Heck.... we never imagined sharing the cable medium with other protocols and the idea of NetWare running over TCP/IP was not even a twinkle (or irritating itch) in our eyes. -- Laura

At this point, we have an Ethernet *frame*.

The NIC Adds Error Detection and Spits the Bits Out

Hey, we might eventually get a packet out on the wire! <puff, puff>

When the Network Interface Card (NIC) receives the frame from the LAN driver, it performs an error checking mechanism on the packet and appends a Cyclical Redundancy Check (CRC) value to the end of the packet typically.

The NIC then sends the packet out as a series of bits -- 1s and 0s.

That's it! Now, let's look at the common formats used by the MAC headers, network-layer header, and transport-layer headers. In this example, I'll use TCP/IP formats -- if you need the IPX/SPX structures, grab a copy of "Novell's Guide to IPX/SPX Analysis," available from Novell Press.

The Ethernet Header Formats

The Ethernet II frame type is the de facto standard frame type used for IP datagram transmissions over Ethernet networks. The Ethernet 802.3 frame type (that includes the 802.2 layer) is the default frame type for IPX/SPX communications.

For information on the 802.2 structure or 802.3 structure, refer to www.ieee.org *-- Laura*

We'll use the Ethernet II frame type for our example of a MAC layer packet format.

As mentioned earlier, the minimum Ethernet frame size is 64 bytes. The maximum Ethernet frame size is 1518 bytes. If a packet does not meet the minimum frame size of 64 bytes, the driver must pad the data field.

For more information on Ethernet technology, visit Charles Spurgeon's web site www.ots.utexas.edu/ethernet/ethernet.html. *To order the IEEE 802.3 CSMA/CD specification, visit http://standards.ieee.org/catalog/IEEE802.3.html. To view common questions regarding the IEEE 802.3 standard, visit the IEEE 802.3 Interpretations page at http://grouper.ieee.org/groups/802/3/interp/. -- Laura*

Figure 4-6 shows the layout of the Ethernet II frame. The very first field shown is the Preamble field. This field is used to provide timing so receiving devices can determine the bit boundaries - this enables the receiver to interpret 1s and 0s correctly.

Analyzers typically don't show you the preamble because it is stripped off by the receiving NIC. Some of the really hot high priced analyzers come with a special NIC that enables you to transmit faulty preamble values to test other devices on the network.

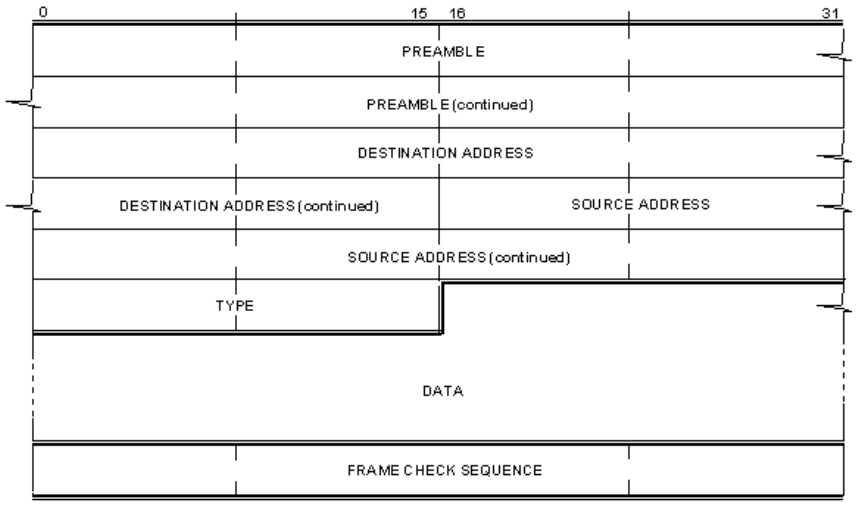


Figure 4-5: The Ethernet II frame format.

The Token Ring Header Formats

Token Ring networking is defined by the IEEE 802.5 standard. Token Ring networks rely on a physical star design although they use a logical ring transmission path.

On a Token Ring network, each Token Ring workstation acts as a repeater - repeating each packet received back out onto the network. If the frame was destined for the receiving Token Ring workstation, the station makes a copy of the frame as it repeats the frame back onto the network.

In the same manner that Ethernet NICs perform an error checking routine on the contents of transmitted packets, Token Ring frames also include a Frame Check Sequence field.

Cisco has some excellent documentation on Token Ring technology. View its site at www.cisco.com. *Order the IEEE 802.5 specification online at* shop.ieee.org/store/product.asp?prodno=SS94583. *-- Laura*

There are two variations of Token Ring frame: Token Ring 802.2 frames and Token Ring SNAP frames. We'll focus on the Token Ring 802.2 frame type.

Figure 4-7 shows the Token Ring 802.2 frame structure.

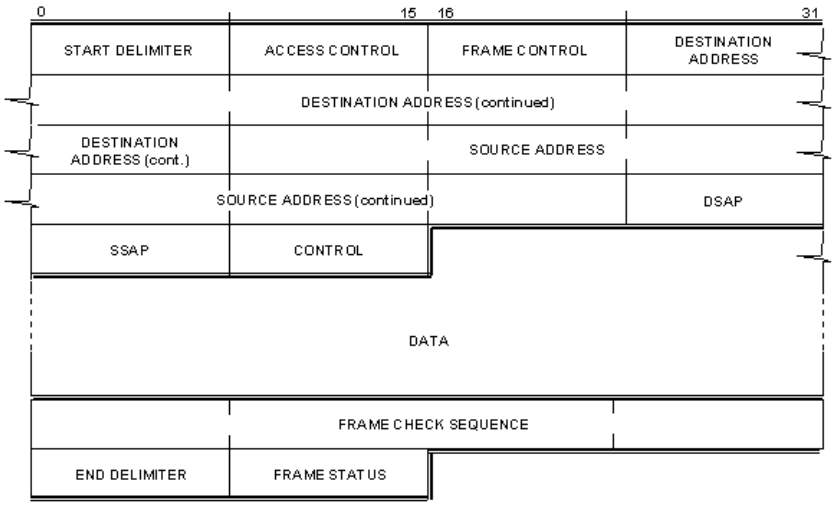


Figure 4-7: The Token Ring 802.2 frame structure.

Ok... enough of this MAC-layer stuff....it's time to move up to the network layer and check out the IP header structure.

The IP Header Format

The RFCs contain the complete details on TCP/IP header structures. In this section, however, I'll provide you with some information on the Internet Protocol (IP), User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) header structures.

Figure 4-8 depicts the entire IP header structure. This section details the header fields and their functions. For more details on each field, refer to RFC 791.

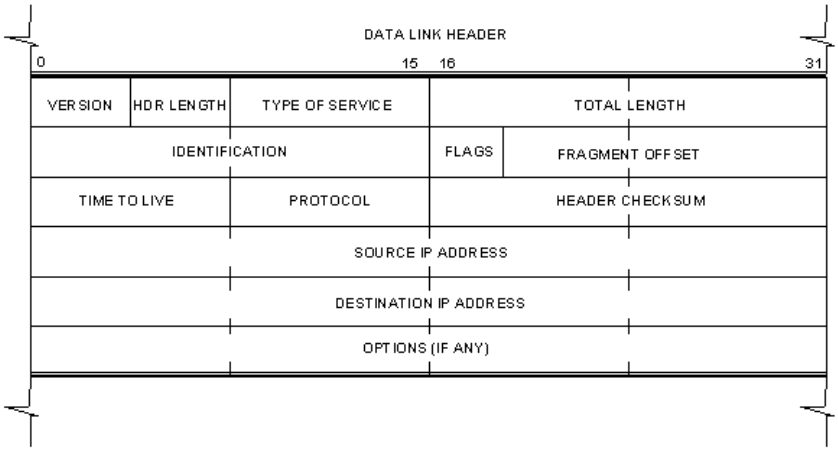


Figure 4-8: The IP header structure.

This next section provides details on the fields contained in the IP header.

Version Field

The first field in the IP header is the version field. We are currently at version 4 - version 6 is in development.

Header Length Field

This field is also referred to as the "Internet Header Length" field or IHL. This field denotes the length of the IP header only - just the IP header. This is necessary because the IP header can support options and therefore, may be varying lengths.

This field value is provided in multiples of four bytes… for example, the actual decimal decode is 5. The analyzers multiply that value by four bytes to come up with the true IP header length value of 20

bytes. Because the Options are rarely used - you can count on 20 byte IP headers as the typical size IP header.

Type of Service Field

As mentioned earlier, the Type of Service (TOS) field actually has two components: Precedence and Type of Service. Precedence is defined in the first three bits and may be used by routers to prioritize traffic that goes through router queues. Type of Service is defined in the next four bits and should be used by routers to follow a specified path type. The last bit of the entire TOS field is reserved and set at zero.

Refer to RFC 791 for more details on the TOS and Precedence bits. -- Laura

The following is a list of the possible settings in the Precedence bits:

Binary	Decimal	Purpose
111	7	Network Control
110	6	Internetwork Control
101	5	CRITIC/ECP
100	4	Flash Override
011	3	Flash
010	2	Immediate
001	1	Priority
000	0	Routine

When you are converting these binary values to decimal, the bits have the following decimal values:

Bit 2	Bit 1	Bit 0	
4	2	1	decimal value

For example, the binary value 111 is decimal value 7 (4 + 2 + 1).

The following is a list of the possible settings in the Type of Service bits:

Binary	Purpose
0000	Default (no specific route defined)
0001	Minimize Monetary Cost
0010	Maximize Reliability
0100	Maximize Throughput
1000	Minimize Delay
1111	Maximize Security

For example, a packet with the binary value 00010000 in the TOS field requires routine service (000) through the lowest delay path (1000).

Some analyzers only show the four basic service types (default, delay, throughput, and reliability).

RFC 2474, "Definition of Differentiated Service Field (DS Field) in the IPv4 and IPv6 Headers" recommends a complete redefinition of the TOS field values and functions. -- Laura

Most likely you will find this field set at 00000000 (default).

Total Length Field

This field defines the length of the IP header and any valid data (does not include any data-link padding). In the example shown in Figure 4-4, the total length is 64 bytes. The first 20 bytes of that are the IP header - this indicates that the remaining packet length is 46 bytes.

Identification Field

Each individual packet is given a unique ID value when it is sent. If the packet must be fragmented to fit on a network that supports a smaller packet size, the same ID number is placed in each fragment. This helps identify fragments that are part of the same 'set' of data.

Flags Field

The Flags field is actually three bits in length and has the following bit value assignments:

Bit 0: Reserved - set to 0.

Bit 1: The Don't Fragment Bit (0=may fragment; 1=don't fragment)

Bit 2: The More Fragments Bit (0=last fragment; 1=more to come)

Typically, fragmentation is allowed. An application may for some reason decide not to allow fragmentation. If so, it sets the Don't Fragment bit to 1.

If fragmentation is allowed and a packet must be fragmented to cross a network that supports a smaller Maximum Transmission Unit

(MTU), the Don't Fragment bit would be set to 0. When the packet is split into multiple fragments (three fragments, for example) the first and second fragments have the More Fragments bit set to 1. The last fragment has the More Fragments bit set to 0 to indicate that it is the final fragment in the set.

Is fragmentation a good thing? Sometimes. Of course, if you have mixed media types (Ethernet and Token Ring, for example), you may need fragmentation to get that big 4,096 byte packet through the 1518-byte Ethernet network by splitting it up into multiple packets. Fragmentation does take time and extra overhead however.

Fragment Offset Field

If the packet is a fragment, this field shows where to place this packet's data when the fragments are being reassembled into a single packet again (at the destination IP host).

This field gives the offset in eight-byte values. For example, the first fragment may have an offset of 0 and contain 1400 bytes of data (not including any headers). The second fragment would have offset value 175 (175 x 8 = 1400).

This field is only in use if the packet is a fragment.

Time to Live (TTL) Field

This field denotes the remaining lifetime (defined as seconds, but implemented as hops through routers) of the packet. Typical starting TTL values are 32, 64, and 128.

Protocol Field

Headers should have some field that defines what is coming up next. For example, in a TCP/IP packet, an Ethernet header should have a protocol identification field (the Type or Ether Type field) to indicate that IP is coming up next. The IP header, likewise, has a Protocol field to indicate what is coming up next. The more commonly seen values in the protocol field are listed below:

#	**Description**
1	Internet Control Message Protocol (ICMP)
2	Internet Group Management Protocol (IGMP)
6	Transmission Control Protocol (TCP)
8	Exterior Gateway Protocol (EGP)
9	Any private interior gateway, such as Cisco's IGRP

17 User Datagram Protocol (UDP)
45 Inter-Domain Routing Protocol (IDRP)
88 Cisco EIGRP
89 Open Shortest Path First (OSPF)

The values 134-254 are unassigned and 255 is reserved by IANA. To obtain the most current list of Protocol field values, visit www.isi.edu/in-notes/iana/assignments/protocol-numbers.

Header Checksum Field

The IP header checksum field provides error detection on the contents of the IP header only - this field does not cover the contents of the packet other than the IP header. This checksum does not include the checksum field itself in the calculation, of course.

This is an error detection mechanism in addition to the data-link error detection mechanism (such as the Ethernet CRC). This additional checking mechanism is required for packets that pass through routers. For example, when an Ethernet packet arrives at a router, the router performs the data-link CRC to ensure the packet has not been corrupted along the way.

After the packet passes the CRC check and is considered 'good,' the router strips off the data-link header - leaving behind an unencapsulated network-layer packet. If the packet did not have any error detection process in place, a faulty router could alter the data and then apply a new data-link header (with a new CRC on the invalid packet) and send the packet on. This network-layer error checking mechanism is required to check router packet corruption.

Source Address Field

This is the IP address of the IP host that sent the packet. In some cases, such as during the DHCP bootup process, the IP host may not know its IP address, so it may use 0.0.0.0 in this field.

This field cannot contain a multicast or broadcast address.

Destination Address Field

This field can include a unicast, multicast or broadcast address. This is the final destination of the packet.

Options Fields

The IP header can be extended by a number of options (although these options are not often used). If the header is extended with options, those options must end on a four-byte boundary because the Internet Header Length (IHL) field defines the header length in four-byte boundaries.

The following list only displays a partial set of options. For the complete list, refer to www.iana.org.

#	Name
0	End of Options List
3	Loose Source Route
4	Time Stamp
7	Record Route
9	Strict Source Route

I could go on all day -- if you've ever attended one of my lectures, you know that to be true. Sigh. Again, for more information on IP, refer to RFC 791.

The UDP Header Structure

When the Protocol field of an IP header contains the value 17 (0x11), the UDP header follows the IP header. The UDP header is short and simple -- only 8 bytes long. The UDP header's main function is to define the process or application that is using the IP and UDP network and transport layers. Figure 4-9 shows the layout of the UDP header.

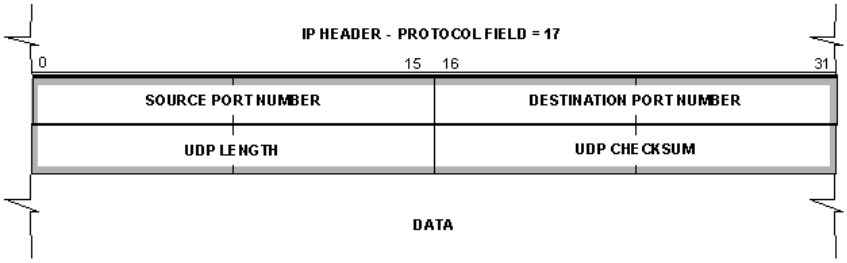


Figure 4-9: The UDP header structure.

UDP is defined in RFC 768. The UDP header only contains four fields:

- Source Port field
- Destination Port field
- Length field
- Checksum field

In this next section we examine the field values and functions of the UDP header.

Source Port Field

The source port field defines the application or process that is sending the packet using the UDP transport. In some cases, a temporary port number is used.

Port numbers are defined into three ranges: the well-known port numbers, the registered port numbers and the dynamic port numbers.

Well-Known Port Numbers (0 through 1023)

Well-known port numbers are assigned to the key or core services offered by systems. well-known port numbers are between 0 and

1023. Some of the more commonly seen well-known port numbers are listed below.

UDP Port	Application or Process
67	Bootstrap Protocol Server (also DHCP Server)
68	Bootstrap Protocol Client (also DHCP Client)
161	Simple Network Management Protocol (SNMP)
162	Simple Network Management Protocol (SNMP) Trap
520	Routing Information Protocol (RIP)

Until 1992, the well-known port number range was between 1 and 255.

Registered Port Numbers (1024 through 49151)

Registered port numbers are assigned to industry applications and processes such as 1433, which is assigned to the Microsoft SQL Server process. Some TCP/IP systems use between 1024 and 5000 for temporary port numbers, although IANA includes that range as part of their Dynamic Port Numbers list, defined next.

Dynamic Port Numbers (49152 through 65535).

Dynamic ports (also referred to as ephemeral port numbers or temporary port numbers). These ports are used as a temporary port for a specific communication.

In most cases, an application or process can use the same number whether it is running over UDP or TCP because the UDP and TCP port number assigned is the same. In some rare cases, however, the UDP and TCP port numbers assigned do not support the same service -- UDP port 520 is a perfect example. The UDP port number 520 is assigned to Router Information Protocol (RIP). The TCP port 520 is assigned to the extended file name server (EFS) process.

Refer to www.iana.org for the complete list of the assigned port numbers and a definition of the three types of port numbers found in UDP and TCP headers. In fact, just bookmark www.iana.org -- it's your best reference for up-to-date field values for TCP/IP communications. -- Laura

Destination Port Field

This field value defines the destination application or process that is using the IP and TCP headers. In some instances the source and destination port numbers are the same for client and server processes. In other instances, you may find a separate and unique number for the client and server process (as in the case of DHCP). Still another variation is to allow the client to use dynamic port numbers for their side of the communications and a well-known port numbers or registered port numbers for the server side of the communications.

Length Field

The length field defines the length of the packet from the UDP header to the end of valid data (not including any data link padding, if padding is required). This field provides a redundant measurement since this information can be determined using the IP header value and using the knowledge that the UDP header is 8 bytes long.

Checksum Field

The UDP checksum is optional. If a checksum is used, the checksum calculation is performed on the contents of the UDP header (except the checksum field itself), the data and a pseudo-header that is derived from the IP header. The UDP pseudo header is not actually contained in the packet -- the UDP pseudo-header is only used to calculate the UDP header checksum and associate the UDP header with the IP header. This pseudo-header is made up from the IP header source address field, the destination address field, an unused field (set to 0), the protocol field and UDP length field.

UDP is pretty simple. The problem with UDP communications, of course is their connectionless nature. There's no guarantee that UDP packets will arrive at a destination. Applications therefore need to have some error detection and connection management capability built in.

TCP Header Structure

Now we examine the TCP header structure shown in Figure 4-10. You should recognize some characteristics of the TCP header, such as the source and destination port values. Now let's examine the other aspects of the TCP header structure, as defined in RFC 793.

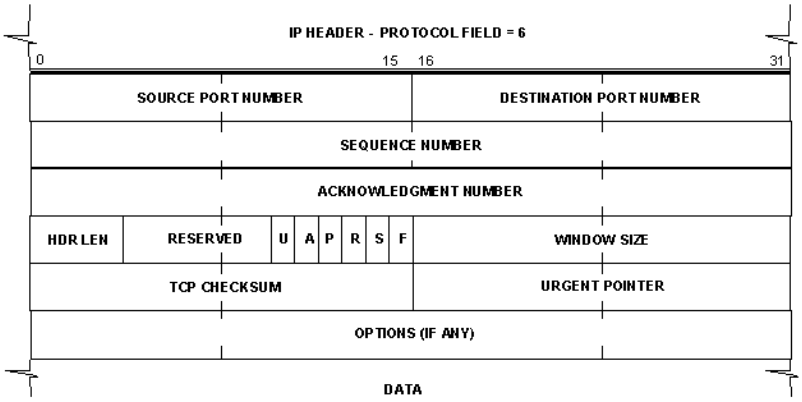


Figure 4-10: The TCP header structure.

Source Port Field

See UDP Source Port field definition in the previous section.

Destination Port Field

See UDP Destination Field definition in the previous section.

Sequence Number Field

This field contains a number that uniquely identifies the TCP segment. This sequence number provides an identifier that enables TCP receivers to identify when parts of a communication stream are missing.

Each TCP Host self-assigns its own sequence number.

Acknowledgment Number Field

The Acknowledgment Number field indicates the next expected sequence number from the other side of the communications.

Data Offset Field

This field defines the length of the TCP header in 4 byte increments. A value of 5 in this field indicates that the TCP header is 20 bytes long. We need this field because the TCP header length can vary depending on the TCP header options used.

While the UDP option field is rarely used, the TCP option field is almost always used during the TCP connection setup to establish the maximum amount of data that can be placed after a TCP header. Hence the need for a field to tell us how long the header is. -- Laura

Flags Field

The following list describes the flags used in the TCP header:

URG (Urgent) Indicates Urgent Pointer field should be examined. If this flag is set, the Urgent Pointer field tells the receiver where they should start reading bytes contained in the data portion of the packet.

ACK (Acknowledgment) The Acknowledgment Number field is relevant. This indicates that the Acknowledgment number indicates the next expected sequence number from the other TCP peer.

PSH (Push) Bypass buffering and pass data straight to upper layer. This is used for time-critical or single-stroke applications. Upon receipt of a packet with the Push flag set, the receiver must not buffer the data - they must pass it directly up to the Application Layer protocol.

RST (Reset) Close the connection. This is used to shut down the connection entirely. This flag is also used to refuse a connection for whatever reason.

SYN (Synchronize) Synchronize sequence numbers - handshake process. This flag is used during the handshake process to indicate that the sender is notify-

	ing the TCP peer of its Sequence Number value.
FIN (Finish)	The transaction is finished This flag is used to indicate that the host has completed a transaction. The FIN flag itself does not explicitly close the connection. If, however, both peers send TCP FIN packets and respond with appropriate ACKs, the connection will be closed.

Window Field

This field indicates the size of the TCP receiver buffer in bytes. A window size of 0 indicates that a sender should stop transmitting - the receiver's TCP buffer is full.

Checksum Field

This TCP checksum is a bit strange, just like the UDP checksum. The checksum is performed on the contents of the TCP header and data (not including data link padding) as well as a pseudo header derived from the IP header. You can get more information on the TCP pseudo-header from RFC 793.

Urgent Pointer Field

This field is only relevant if the URG pointer is set. If the URG pointer is set, the receiver must examine this field to see where to look/read first in the packet. RFC 1122 interprets the Urgent Pointer field value differently than RFC 793. Windows 2000 can be configured to interpret the Urgent Pointer field according to RFC 1122, if desired.

TCP Options Field(s)

This option field is… well, optional. One option you will certainly see in use is the Maximum Segment Size (MSS) option - it is used in the first two packets of the three-way handshake process. The purpose of this option is to define what segment size of the hosts supports. The hosts will use the lowest common denominator between the two MSS values.

The complete TCP Options list can be found online at www.isi.edu/in-notes/iana/assignments/tcp-parameters.

Walking Through a Trace

It always helps to have someone walk you through a trace so you become familiar with how an application or process works. In this example, I'll walk you through an HTTP communication.

First, however, let's just provide some basic background information on how HTTP works. Much of this information is from the HTTP specifications available at www.ietf.org.

Hypertext Transfer Protocol (HTTP) is an application-level protocol that provides distributed information from various hypermedia systems. HTTP has been in use on the Web since 1990. Currently, HTTP v1.0 (documented in RFC 1945) is widely-implemented, but HTTP v1.1 is a Proposed Standard (documented in RFC 2616).

HTTP uses a request/response model, where HTTP clients send a request to the server and expect a reply packet that contains a status code. Requests include a Uniform Resource Identifier (URI) that identifies the desired web resource. By default, HTTP communications use port 80. In this section, we define the basic elements that are common to HTTP versions 1.0 and 1.1.

The HTTP Elements

HTTP clients send HTTP requests that contain a method, or command, that indicates what the client wants. The requests also contain a URI indicating the desired resource. HTTP servers respond with a numeric code (the status code) and data (if appropriate). Figure 4-11 indicates how these elements work together to transfer HTTP data.

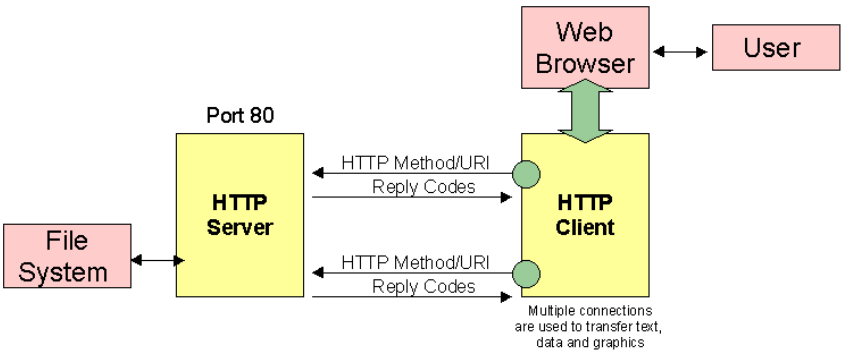


Figure 4-11: The HTTP communications architecture.

Uniform Resource Identifiers

URIs are also referred to as web addresses and a combination of Universal Resource Locators (URL) and Universal Resource Names (URN). In accordance with RFC 1945, "Uniform Resource Identifiers are simply formatted strings which identify--via name, location, or any other characteristic--a resource."

HTTP Methods

HTTP methods are commands that the HTTP client issues to the HTTP server. HTTP v1.0 only defined three methods - HTTP v1.1 expanded the list as shown in the following table.

Methods	Description
GET	Used to retrieve the information that is identified by the URI contained in the request.
HEAD	Used to retrieve the meta information related to the desired URI.
POST	Used to send data to the HTTP server. The data should be placed as a new subordinate of the resource identified by the URI in the request.
OPTIONS	Used to determine the options and/or requirements associated with a resource or the capabilities of a server.
PUT	Used to send data to the HTTP server. The data should be stored in the URI indicated in the POST request.
DELETE	Used to delete the resource defined by the URI in the DELETE request.
TRACE	Used to invoke a remote, application-layer loop-back of the request message, allowing the client to see what the server is receiving from the client.
CONNECT	Used with a proxy that can dynamically switch to being a tunnel (e.g. SSL tunneling)

Status Codes

Status codes are sent from the HTTP server to indicate success or failure of the request. The HTTP status codes defined in HTTP v1.1 are listed in the next table.

Code	Definition
1xx	Informational
100	Continue
101	Switching Protocols
2xx	Successful
200	OK
201	Created
202	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content
3xx	Redirection 3xx
300	Multiple Choices
301	Moved Permanently
302	Found
303	See Other
304	Not Modified
305	Use Proxy
306	(Unused)
307	Temporary Redirect
4xx	Client Error 4xx
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request-URI Too Long
415	Unsupported Media Type
416	Requested Range Not Satisfiable
417	Expectation Failed
5xx	Server Errors

500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout
505	HTTP Version Not Supported

Sample HTTP Communications

HTTP commands and status codes follow immediately after the TCP header, as shown in Figure 4-12.

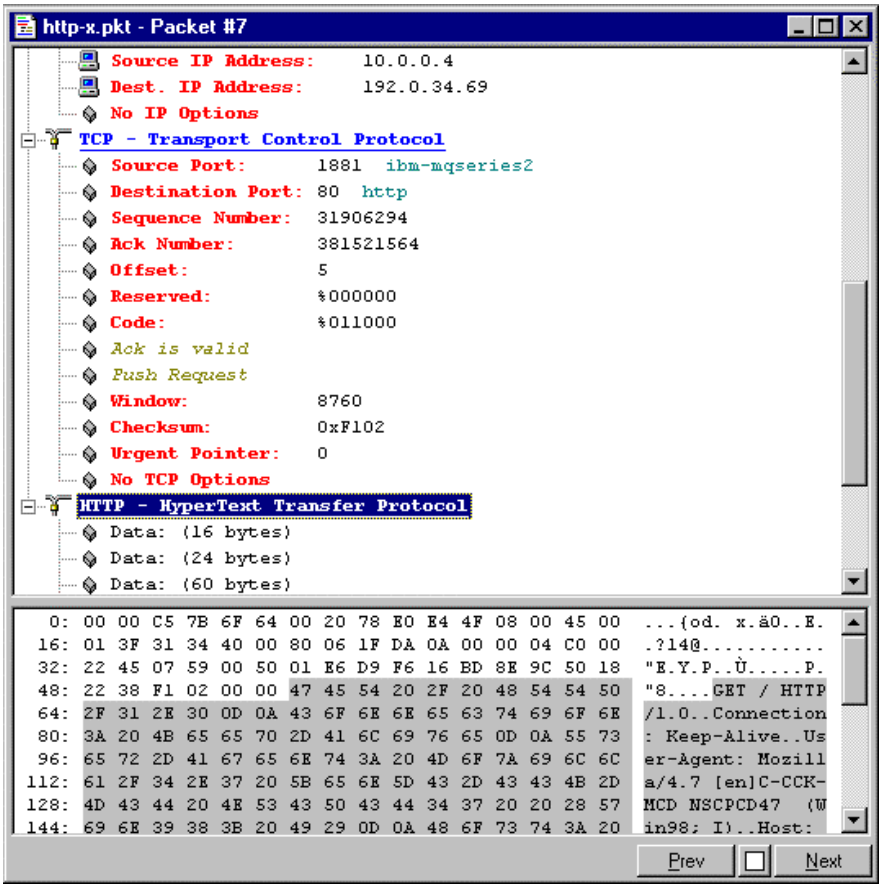

http-x.pkt - Packet #7
Source IP Address: 10.0.0.4
Dest. IP Address: 192.0.34.69
No IP Options
TCP - Transport Control Protocol
Source Port: 1881 ibm-mqseries2
Destination Port: 80 http
Sequence Number: 31906294
Ack Number: 381521564
Offset: 5
Reserved: %000000
Code: %011000
Ack is valid
Push Request
Window: 8760
Checksum: 0xF102
Urgent Pointer: 0
No TCP Options
HTTP - HyperText Transfer Protocol
Data: (16 bytes)
Data: (24 bytes)
Data: (60 bytes)
0: 00 00 C5 7B 6F 64 00 20 78 E0 E4 4F 08 00 45 00 ...{od. x.äO..E.
16: 01 3F 31 34 40 00 80 06 1F DA 0A 00 00 04 C0 00 .?14@.....
32: 22 45 07 59 00 50 01 E6 D9 F6 16 BD 8E 9C 50 18 "E.Y.P..Ù.....P.
48: 22 38 F1 02 00 00 47 45 54 20 2F 20 48 54 54 50 "8....GET / HTTP
64: 2F 31 2E 30 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E /1.0..Connection
80: 3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 55 73 : Keep-Alive..Us
96: 65 72 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C er-Agent: Mozill
112: 61 2F 34 2E 37 20 5B 65 6E 5D 43 2D 43 43 4B 2D a/4.7 [en]C-CCK-
128: 4D 43 44 20 4E 53 43 50 43 44 34 37 20 20 28 57 MCD NSCPCD47 (W
144: 69 6E 39 38 3B 20 49 29 0D 0A 48 6F 73 74 3A 20 in98; I)..Host:
Prev
Next

Figure 4-12: A nicely decoded HTTP GET request.

Now let's examine the various steps involved with browsing a web site (www.iana.org) using HTTP. In this communication, "S" represents the HTTP server and "C" represents the HTTP client.

This trace file is called *http-x.pkt* -- it is available on the www.packet-level.com website.

Process	Packet Sequences	
Establish TCP Connection (the three-way TCP handshake) Server port 80 Client port 1881	C > S S > C C > S	TCP SYN TCP SYN ACK TCP ACK
Get Web Page Send Client Attributes Server Attributes	C > S	HTTP Get Request (URL: www.iana.org) Provide Connection, User-Agent, Host, Accept, Accept-Encoding, Accept-Language and Accept-Charset.
	S > C	HTTP Get Reply; Code 200 Provide Time, Server Type, Last Modified Date/Time, Etag, Accept-Ranges, Content-Length, Keep-Alive, Content-Type.
		Start web page download Continue this connection to get the web page while the client opens additional connections to get the graphic images.
Establish 2nd TCP Connection (the three-way TCP handshake) Server port 80 Client port 1882	C > S C > S C > S	TCP SYN TCP SYN ACK TCP SYN
Get First Graphic Page Send Client Attributes Server Attributes	C > S	HTTP Get iana1.jpg Provide same client information as well as referrer (the page that listed the graphic file that the client wants to get.
	S > C	HTTP Get Reply; Code 200

		Provide same attribute information. Start sending iana1.jpg. Continue this connection to get the graphic while the client opens another connection to get the next graphic image. Close the connection when the graphic has downloaded successfully.
Establish 3rd TCP Connection (the three-way TCP handshake) Server port 80 Client port 1883	C > S S > C C > S	TCP SYN TCP SYN ACK TCP ACK
Get First Graphic Page Send Client Attributes Server Attributes	C > S	HTTP Get blueball.gif Provide same client information as well as referrer (the page that listed the graphic file that the client wants to get.
	S > C	HTTP Get Reply; Code 200 Provide same attribute information. Start sending blueball.gif. Continue this connection to get the graphic until the entire image has been downloaded to the client. Close the connection when the graphic has downloaded successfully.

The communications shown above continues along the same general pattern - open a TCP connection, send a GET request for a graphic, receive a file and finally, close the connection when the graphic file is downloaded.

At the time of this book revision, I placed three more 'walk throughs' online at www. packet-level.com*: Telnet, FTP and SMTP. Check them out! -- Laura*

This chapter should make have made you more comfortable with the frame formats and general interpretation of the fields of IP, UDP and TCP header structures. I know, I know... I didn't really go in-depth with the Ethernet or Token Ring headers, but Ethernet is so easy and Token Ring takes too long. Token Ring frame structures and functions are covered in "Novell's Guide to IPX/SPX Analysis."

Now... on to the chapter test.

Chapter 3 Test

Here we go. These questions are focused on packet interpretation and understanding.

3.1 A TCP/IP packet is addressed to the destination port number 53. What can you assume about this packet?

3.2 You have noticed a TCP communication that uses a Window field setting of 0. What does this indicate?

3.3 You are interested in capturing only ARP traffic on a network. What Ethernet type value should you filter on?

3.4 OK... this one may take a couple of minutes... You are working with an analyzer that does not decode many packet types. Ugh. The following graphic provides an example of an undecoded packet. What can you figure out using the header structure information and sample headers shown in this packet?

```
 0: 00 00 C5 7B 6F 64 00 20 78 E0 E4 4F 08 00 45 00  ...{od. x.ä0..E.
16: 00 28 5F 4C 40 00 80 06 19 97 C0 A8 00 43 C0 A8  .(_L@........C..
32: 00 59 09 A0 00 50 0D A1 B5 23 41 1B E4 34 50 10  .Y...P..µ#A.ä4P.
48: 22 38 16 1C 00 00 00 00 00 00 00 00 6A D6 8C 48  "8..........j..H
```

CHAPTER 4

Identifying Typical Problems

This chapter is the one that you immediate gratification folks will jump to. It lists the most common problems that I see on Ethernet, Token Ring, IPX/SPX and TCP/IP networks. It should give you a jump-start on your troubleshooting by letting you know what to start looking for.

If you did just jump to this chapter, consider spending some time back on Chapter 2, "How Do You Start?" That chapter gives a list of analysis tasks that you can do today - right now. You can use those tasks to move you towards actually troubleshooting.

This chapter doesn't include all the possible problems that you may encounter on your network - sound depressing? Well, don't let it be. Consider this chapter your checklist of possible problems to prepare for and consider. - Laura

The Elusive 'Typical Problems'

The term 'typical' is simply used as an indication that these problems are some of the more common that occur on networks. In order to avoid being stumped by these problems, you should know what the problems are (and the technology behind them).

Ethernet/802.3 Problems

This section focuses specifically on the standard Ethernet/802.3 problems that may occur on the network. There are many 'flavors' of Ethernet/802.3 implementation, but we'll focus on the most popular ones, 10BaseT and 100BaseT. Naturally, we must start with a quick overview of how CSMA/CD (Carrier Sense Multiple Access/ Collision Detection) functions.

This isn't the ultimate definition of the functionality, but it gives you something to use as a basic reference as you go through this section.

CSMA/CD 101

CSMA/CD communications involves the following elements:

1. Multiple devices have concurrent access to the medium.
2. All devices listen (carrier sense) to the medium before transmitting onto the cable. If the cable is busy, the device must defer (wait) until the cable is free.
3. If the cable is not busy, the device may transmit while it simultaneously listens for involvement in a collision. A collision occurs when two or more packets hit each other (typically head-on). The result of a collision is a fragment.
4. All packets that are transmitted include an error detection mechanism (cyclical redundancy check).
5. If a device's packet is involved in a collision, the device must begin the backoff process. This process involves running an algorithm to obtain a new time to retransmit the packet. When the device retransmits, it again listens to see if its next packet is involved in a collision, also.
6. A device can attempt to retransmit up to 15 times. If the device cannot transmit a packet without collision within the 15 attempts, it registers an "ExcessCollisionsCount." Some driver software will display this information in their statistics reporting process.
7. The minimum packet size is 64 bytes. The maximum packet size is 1518 bytes.
8. There are two key frame types used on CSMA/CD networks -- Ethernet II and 802.3. The de facto industry standard frame type for IP communications is the Ethernet II frame type.

Figure 4-1 shows the decoded Ethernet II frame structure.

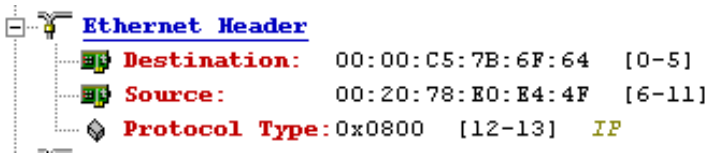


Figure 4-1: The Ethernet II frame structure with offsets shown.

The field offset is shown in brackets to the right of the field values. The ability to turn on offsets is a wonderful option in EtherPeek. -- Laura

Not everyone shares the same naming convention on the frame types. What Novell refers to as the Ethernet_802.2 frame type, Cisco refers to as the *sap* frame type. The following chart should help with the comparison between industry, Cisco and Novell's naming standard for Ethernet frame types:

Industry Name	Cisco Name	Novell Name
No name; illegal frame	novell-ether	Ethernet_802.3
Ethernet II	arpa	Ethernet_II
Ethernet 802.3	sap	Ethernet_802.2
Ethernet SNAP	snap	Ethernet_SNAP

For some details on why you should avoid the Ethernet_802.3 frame, refer to the Greypaper entitled simply "Why You Should Avoid the Ethernet_802.3 Frame Type" at www.packet-level.com*. -Laura*

From a communications standpoint, the Ethernet II frame is the most streamlined, effective frame structure. It is simple and elegant, only requiring a single Type field to identify what comes next. Naturally, however, the functionality afforded by the Logical Link Control (802.2) layer is not available with this frame structure.

Now let's delve into the types of problems that can occur on an Ethernet network. These problems include:

- Collisions
- CRC Errors
- Short Frames
- Long Frames
- Alignment Errors
- Jabber

Let's examine each of these errors to find out what they are caused by and how to isolate the source.

Collisions (Local/Remote/Late)

As you learned earlier, collisions occur when two or more packets run into each other. Most collisions in the CSMA/CD world are head-on collisions.

The result of a collision is a fragment (a packet that is less than 64 bytes with a bad CRC). There are three types of collisions - they are each unique communication patterns, but are often lumped together by analyzers.

- *Local Collision*: A local collision is one that occurs on the local side of a repeating device (such as a hub). Local collisions are identified by a signal that exceeds that of two or more simultaneous transmissions (a 'spike') along with a fragment on the wire. Figure 4-2 shows what you see on an analyzer when a collision has occurred.

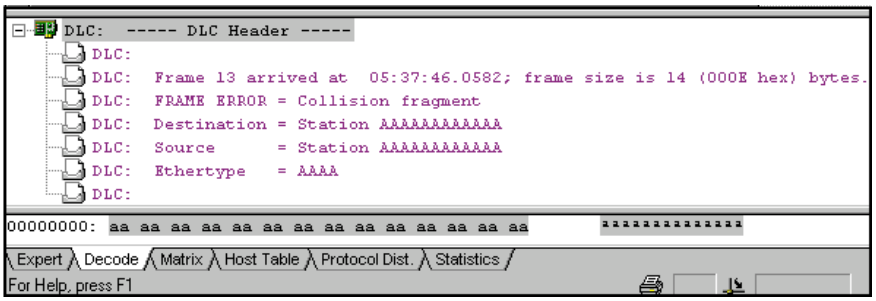

DLC: ----- DLC Header -----
DLC:
DLC: Frame 13 arrived at 05:37:46.0582; frame size is 14 (000E hex) bytes.
DLC: FRAME ERROR = Collision fragment
DLC: Destination = Station AAAAAAAAAAAA
DLC: Source = Station AAAAAAAAAAAA
DLC: Ethertype = AAAA
DLC:
00000000: aa aa aa aa aa aa aa aa aa aa aa aa aa aa
Expert
Decode
Matrix
Host Table
Protocol Dist.
Statistics
For Help, press F1

Figure 4-2: The analyzer depicts the result of a collision.

- *Remote Collision*: A remote collision is one that occurs on the other side of a repeating device (that does not forward the spike in the signal, but does forward the fragment packet).

- *Late Collision*: A late collision is one that is not a head-on collision, but a collision that occurs further into the packet. This type of collision is identified by the fact that the address fields are often readable. Late collisions are an indication of a faulty network interface card.

I've seen late collisions caused by a 'deaf' network interface card. A deaf card acts as if it is sensing the carrier before transmitting, but hey! It's always quiet to a deaf card. A deaf card transmits whenever it wants to - its packets sideswipe any packets cruising by. -Laura

A high number of collisions with a high utilization rate typically indicate a congested network that requires some bandwidth relief. If the utilization is low when the collision count peaks, most likely there is a faulty device on the network.

CRC Errors

Before the Ethernet packet is placed out on the wire, the network interface card runs an algorithm on the contents of the frame. The result of this equation is placed in the Frame Check Sequence field at the end of the frame.

Upon receipt of a frame, a recipient network interface card performs a CRC calculation on the packet (with the exception of the Frame Check Sequence field). It compares its result to the value at the end of the incoming packet. If the two numbers match, the packet is considered valid. If not, the packet is considered corrupt.

Lots of CRC errors from a single source address indicate a possible problem with the NIC or cable to the hub/switch port. Lots of CRC errors from various sources indicate cabling problems.

Short Frames

The IEEE 802.3 specification dictates a minimum packet size of 64 bytes (including the 4-byte CRC). A short frame is less than 64 bytes with a valid CRC. These short frames are typically formed by faulty or out-of-specification LAN drivers.

Long Frames

The IEEE 802.3 specification dictates a maximum packet size of 1518 bytes (including the 4-byte CRC). A long packet is greater than 1518 bytes with a valid CRC. Just like short frames, these long frames are typically formed by faulty or out-of-specification LAN drivers.

Alignment Errors

IEEE 802.3 specifications indicate that all frames must end on an 8-bit boundary. For example, a frame can be 64 bytes, 65 bytes, 66 bytes, etc. Frames cannot be 64 bytes and 3 bits, 65 bytes and 5 bits, and so on. Alignment errors are also CRC errors because the last 4 bytes (with the trailing bits) is an invalid Frame Check Sequence field.

Jabber

In presentations, I often refer to jabber packets as the mother-in-law of all packets. Jabber 'packets' are typically 'streams' of nonsense data. Jabber is a stream of data that exceeds 1518 bytes with an invalid CRC value (sometimes, you'll never even know where the CRC is). Jabber is typically caused by a faulty network interface card.

Troubleshooting Ethernet Today

Now is the time to plug in your analyzer and take a look at the Ethernet errors. If your network is reasonably 'clean' (lacking errors), you may need to run the analyzer overnight to catch some errors. Consider that if you never get any errors, your network interface card may not be a promiscuous mode card.

- What types of errors do you see on your network?
- Can you document your findings in a graphical format?

In the next section, we'll take a brief look at Token Ring troubleshooting.

Token Ring Problems

This section focuses specifically on the standard Token Passing Ring/802.5 problems that may occur on the network. Naturally, we must start with a quick view of how Token Ring functions.

This isn't the ultimate definition of the functionality, but it gives you something to use as a basic reference as you go through this section.

Token Ring 101

On a Token Ring network, a token is passed from one device to another in a counter clockwise direction. When a station has data to transmit, it 'captures' the token and releases a frame that contains data.

Ok, Ok… it doesn't really capture the token. A token is sort of a trimmed down data frame (without the data).

The frame will circulate the entire ring and get stripped off by the original transmitting station. That station must release a token to allow another downstream neighbor (the neighbor located on the side that the tokens and frames travel to next) gets a chance to transmit. Every data frame has bits that can be toggled by the recipient to indicate that it received the frame.

The Token Ring frame structure is quite different than the Ethernet frame structure. Figure 4-3 shows a Token Ring 802.2 frame.

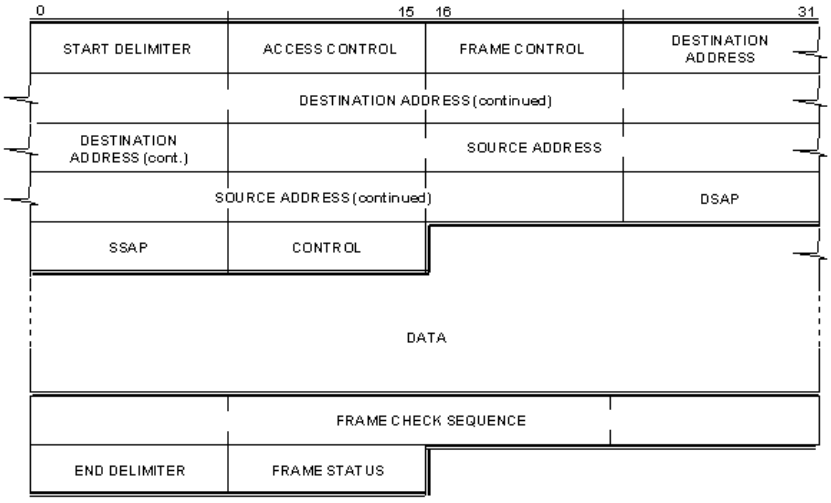


Figure 4-3: The Token Ring 802.2 frame structure.

Token Ring communications include a number of management processes to ensure the integrity of the ring. The following lists the processes involved in creating and keeping a healthy ring:

1. There are two basic Token Ring speeds: 4 Mbps and 16 Mbps. You cannot mix speeds on a single ring. Many Token Ring cards auto-sense the speed before inserting into the ring.
2. All Token Ring devices perform a self-test before entering the ring. The self-test includes a duplicate Media Access Control (MAC) address test.
3. Every ring has one self-assigned Active Monitor. The Active Monitor is responsible for cleaning up a 'dirty' ring and releasing a new token on the ring when required. The Active Monitor also maintains the clock for the ring and ensures that frames do not circle endlessly.
4. Every 7 seconds, the Active Monitor starts a neighbor learning process called the Ring Poll process. Every Token Ring station participates in this process in order to learn the hardware address of its upstream neighbor (the neighbor that will be forwarding frames and tokens to the local device).
5. A Monitor Contention process occurs when the Active Monitor leaves the ring to enable a new Active Monitor to be selected.
6. When a fatal error has occurred on the ring, the Ring Purge process is required to 'clean' the ring before the Active Monitor can release a brand new token.
7. All Token Ring stations periodically report errors to a Ring Error Monitor (MAC address 0xC0-00-00-00-00-08) whether you actually have Ring Error Monitor software installed or not. As an analyst, you want to capture and view all these error reports to determine the health of the ring and isolate trouble spots. If you don't have Ring Error Monitor software installed on your network, the error report frames circulate the ring and are removed by the originating station.
8. Token Ring networks have many self-healing abilities, but a ring can not recover from all errors. If the ring cannot recover from an error (such as an 'open' condition that does not allow a token or frame to circle the ring), the station that notices the open condition will begin to beacon. Beaconing is the process of blaming your upstream neighbor for not sending you tokens or frames. You can intercept these beacons with your analyzer and determine a fault domain (the elements

between the beaconing station and its upstream neighbor). Hopefully, your analyzer was already plugged into the ring, because most network interface cards won't allow you to insert into a beaconing ring.

Figure 4-4 shows one of those MAC frames. As you can see, it defines the error that is being reported. Very nice, eh?

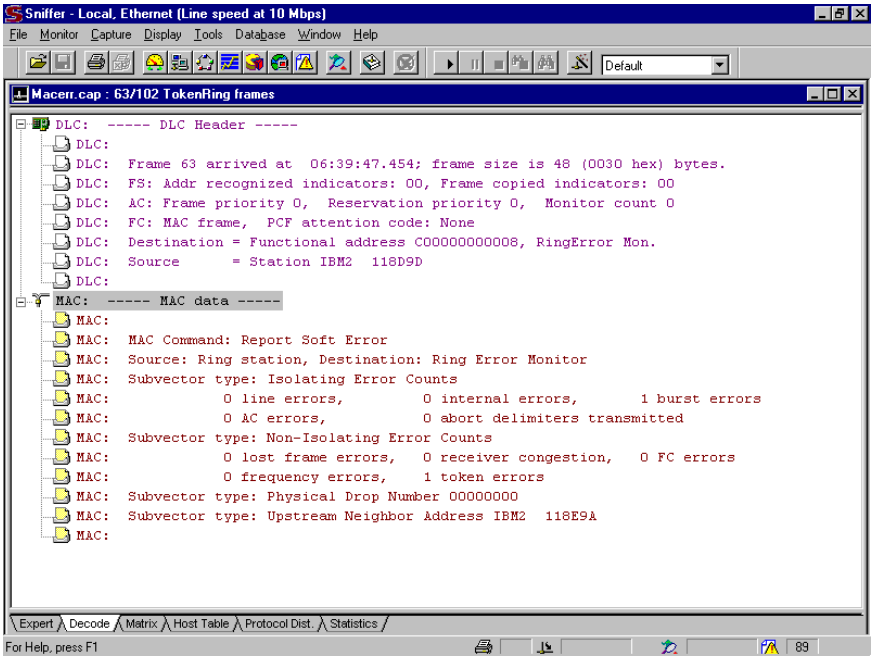

Sniffer - Local, Ethernet (Line speed at 10 Mbps)
File Monitor Capture Display Tools Database Window Help
Default
Macerr.cap : 63/102 TokenRing frames
DLC: ----- DLC Header -----
DLC:
DLC: Frame 63 arrived at 06:39:47.454; frame size is 48 (0030 hex) bytes.
DLC: FS: Addr recognized indicators: 00, Frame copied indicators: 00
DLC: AC: Frame priority 0, Reservation priority 0, Monitor count 0
DLC: FC: MAC frame, PCF attention code: None
DLC: Destination = Functional address C00000000008, RingError Mon.
DLC: Source = Station IBM2 118D9D
DLC:
MAC: ----- MAC data -----
MAC:
MAC: MAC Command: Report Soft Error
MAC: Source: Ring station, Destination: Ring Error Monitor
MAC: Subvector type: Isolating Error Counts
MAC: 0 line errors, 0 internal errors, 1 burst errors
MAC: 0 AC errors, 0 abort delimiters transmitted
MAC: Subvector type: Non-Isolating Error Counts
MAC: 0 lost frame errors, 0 receiver congestion, 0 FC errors
MAC: 0 frequency errors, 1 token errors
MAC: Subvector type: Physical Drop Number 00000000
MAC: Subvector type: Upstream Neighbor Address IBM2 118E9A
MAC:
Expert Decode Matrix Host Table Protocol Dist. Statistics
For Help, press F1
89

Figure 4-4: Packets addressed to the Ring Error Monitor list the ring errors.

Let's examine the types of errors that can occur on a Token Ring network so you'll know what to look for.

Line Errors

This is the Token Ring version of a CRC error and indicates a corrupt frame. If one station reports line errors consistently, that station's network interface card may be failing. If, however, line errors are reported from a number of devices, you most likely have a cable problem.

Internal Errors

Internal errors indicate a problem with the reporting device's network interface card. If the internal error counter increments consistently, replace the card.

Burst Errors

Burst errors are reported when a device notices that the network signal has 'flatlined' or gone without transition for 5 half-bit times. Token Ring communications has timing built into every bit in the form of a mid-bit transition in the signal. The absence of these transitions indicates that the ring is open.

Although burst errors naturally occur during the time when stations enter and exit the ring, you do not want to see a rapidly increasing number of burst errors on the network. If you do, start checking your cables and ports for faults.

ARI/FCI Errors

An ARI (Address Recognized Indicator) and FCI (Frame Copied Indicator) error is reported when a fault occurs during the Ring Poll process. All devices should take turns participating and recognize the last upstream station that communicated. If two ring stations participate consecutively, but the second one does not acknowledge the participation of the first, an ARI/FCI error is reported.

Abort Delimiter

An abort delimiter is reported when a station 'hiccups' on the network. If a station accidentally starts transmitting, it will stop and send an abort sequence, then report an abort delimiter error.

Lost Frame

All Token Ring stations expect to see the frame they transmitted return to them after it circles the ring. If the frame does not return, the station notes a lost frame error.

Receive Congestion

Receive congestion is reported when a station recognizes packets sent to its address, but it cannot copy them into memory because there is no buffer space available. Buffer space may not have been freed up by an application or perhaps a station is being overwhelmed with data from another device. A card with a larger on-board buffer on the device's Token Ring card may solve the problem.

Frame Copied Error

This error is reported when a device notices that someone else has received and copied a frame that was intended for the local device. This indicates that there are possibly two devices with the same MAC address on the network. This error should never occur.

Frequency Error

This error indicates a problem with the ring's clock. Since the Active Monitor is responsible for the clock, the Monitor Contention process can solve this problem by selecting a new Active Monitor.

Token Error

This error can only be reported by the Active Monitor. A token error indicates that a token or frame has become corrupt, a token is lost, or there is a circulating frame. This error forces the Active Monitor to purge the ring and transmit a new token.

Troubleshooting Token Ring Today

Now is the time to plug in your analyzer and take a look at the Token Ring errors. Pay special attention to the management frames that traverse the ring. Filter on any packets addressed to the Ring Error Monitor (address 0xC0-00-00-00-00-08).

If your network is reasonably 'clean' (lacking errors), you may need to run the analyzer overnight to catch some errors.

Remember that if you never get any errors, your network interface card may not be a promiscuous mode card. The tropic chipset (found on the 'True Blue IBM' cards and 'True Blue Compatible' cards) does not perform promiscuous functions - you can only see your own traffic. Make certain that your analyzer does not use this type of chipset.

- What types of errors do you see on your network?
- Can you document your findings in a graphical format?

TCP/IP Problems

TCP/IP is definitely the protocol of choice these days. If you don't have TCP/IP in-house, you're probably getting ready to implement it. This section will give you a brief overview of the TCP/IP protocol elements, the communication process and finally, the common TCP/IP problems to watch for.

Run, don't walk, to www.ietf.org to get RFC 792, "Internet Control Message Protocol" As a network analyst, you must understand this protocol above all others. We also have a certificate course on Packet-Level ICMP available online at www.podbooks.com. Knowing ICMP will improve the quality of your life! -Laura

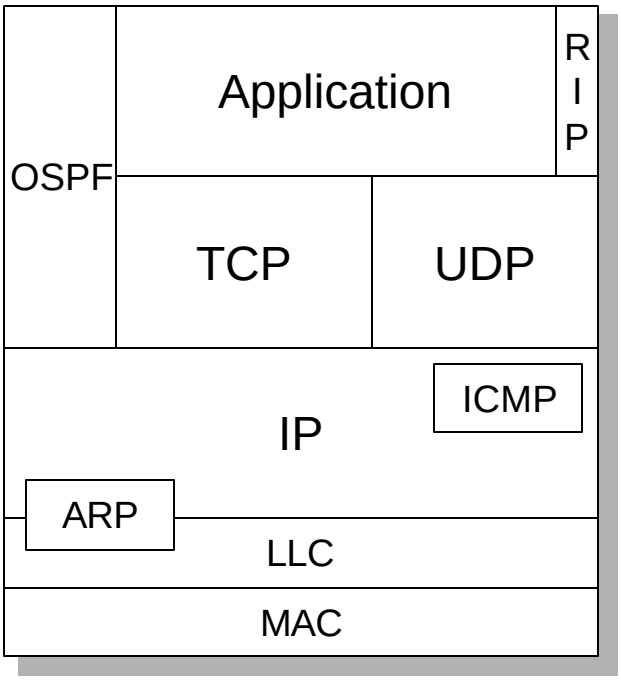


Figure 4-5: The basic TCP/IP stack elements with the lovely ICMP highlighted.

Let's take a quick look at how TCP/IP communications work - we'll do this without focusing on any particular application.

TCP/IP uses a five-step resolution process that is listed below:

1. Resolve the application to the port number. The current TCP/IP assigned port number list is available at www.packet-level.com.
2. Resolve the host name to the IP address. First, you must look at the local host file (if one exists). Next, send queries to the

DNS (Domain Name System) server. If no answer, go to the next DNS server configured on the client. Still no answer? Then you can't get there.

3. Determine if the destination address indicates that the destination device is local or remote. Upon receipt of the IP address, place your own network mask on the destination address and compare the network portion of your address to the desired target.
4. If local, resolve the MAC address of the local target. Check ARP (Address Resolution Protocol) cache for the information. If it does not exist, send an ARP broadcast looking for the target's hardware address.
5. If remote, perform route resolution to identify the appropriate 'next-hop' router. Look in your local routing tables to determine if you have a host or network router entry for the target. If neither entry is available, check for a default gateway entry.
6. If remote, resolve the MAC address of the next-hop router. Check your ARP cache first. If the information does not exist in cache, send an ARP broadcast out to get the information.

This process is illustrated in Figure 4-6.

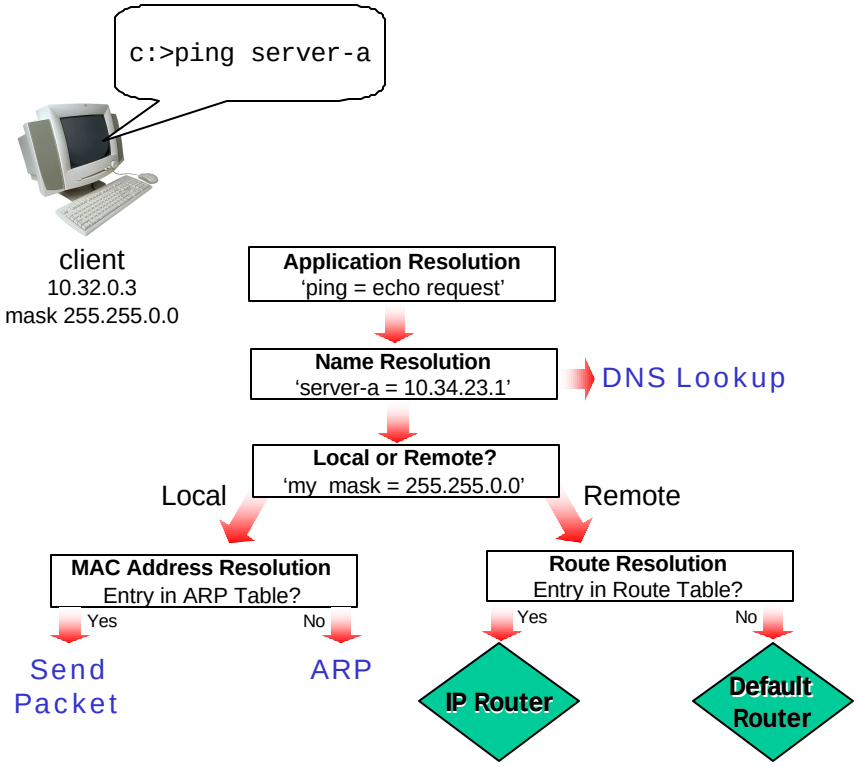


Figure 4-6: The basic resolution process in TCP/IP.

Well, now that you've seen the basic communications process of TCP/IP, let's take a look at the places where things can go wrong. Although this is not an all-inclusive list, it will give you a nice start and hints to check out your own network traffic. First, however, we must take a moment to focus on ICMP. This protocol will be one of your first packet filters set when you troubleshoot a TCP/IP network.

Loving, Living, Eating and Breathing ICMP

ICMP is my favorite protocol in the world these days. I use it to troubleshoot network problems and examine network designs. The following list defines the types of ICMP messages that can be sent on the network.

This list is based on IANA documentation. To obtain the most current version of this list, access www.iana.org.

Type	Name
0	Echo Reply [RFC 792]
1	Unassigned
2	Unassigned
3	Destination Unreachable [RFC 792]
4	Source Quench [RFC 792]
5	Redirect [RFC 792]
6	Alternate Host Address
7	Unassigned
8	Echo [RFC 792]
9	Router Advertisement [RFC 1256]
10	Router Selection [RFC 1256]
11	Time Exceeded [RFC 792]
12	Parameter Problem [RFC 792]
13	Timestamp [RFC 792]
14	Timestamp Reply [RFC 792]
15	Information Request [RFC 792]
16	Information Reply [RFC 792]
17	Address Mask Request [RFC 950]
18	Address Mask Reply [RFC 950]
19	Reserved (for Security)
20-29	Reserved (for Robustness Experiment)
30	Traceroute [RFC 1393]
31	Datagram Conversion Error [RFC 1475]
32	Mobile Host Redirect
33	IPv6 Where-Are-You
34	IPv6 I-Am-Here
35	Mobile Registration Request
36	Mobile Registration Reply
37	Domain Name Request
38	Domain Name Reply
39	SKIP
40	Photuris
41-255	Reserved [JBP]

The initials JBP indicate that an assignment was made by Jon B. Postel. Jon Postel was one of the founders of the Internet protocol suite. With a long beard and an intensely brilliant mind, he helped shape the communications system of the Internet and millions of private networks until his untimely death in October 1998. You can learn more about this luminary at www.iana.org/postel/index.html*. -Laura*

Many of these ICMP types have a "code" field. Here are the types that use code fields and their assigned code field values.

Type Name

3 Destination Unreachable

Codes

0 Net Unreachable
1 Host Unreachable
2 Protocol Unreachable
3 Port Unreachable
4 Fragmentation Needed and Don't Fragment was Set
5 Source Route Failed
6 Destination Network Unknown
7 Destination Host Unknown
8 Source Host Isolated
9 Communication with Destination Network is Administratively Prohibited
10 Communication with Destination Host is Administratively Prohibited
11 Destination Network Unreachable for Type of Service
12 Destination Host Unreachable for Type of Service
13 Communication Administratively Prohibited
14 Host Precedence Violation
15 Precedence cutoff in effect

5 Redirect

Codes

0 Redirect Datagram for the Network (or subnet)
1 Redirect Datagram for the Host
2 Redirect Datagram for the Type of Service and Network
3 Redirect Datagram for the Type of Service and Host

6 Alternate Host Address

Codes

0 Alternate Address for Host

11 Time Exceeded

Codes

0 Time to Live exceeded in Transit
1 Fragment Reassembly Time Exceeded

12 Parameter Problem

Codes

0 Pointer indicates the error
1 Missing a Required Option
2 Bad Length

40 Photuris

Code

0 Reserved
1 unknown security parameters index
2 valid security parameters, but authentication failed
3 valid security parameters, but decryption failed

Most of these codes are self-explanatory - get RFC 792 for the details. You can also see some details in the online number assignment listing at www.iana.org.

Figure 4-7 provides an example of an ICMP packet - I've compressed the Ethernet and IP headers. What is this ICMP packet telling you? (See below the graphic for the answer.)

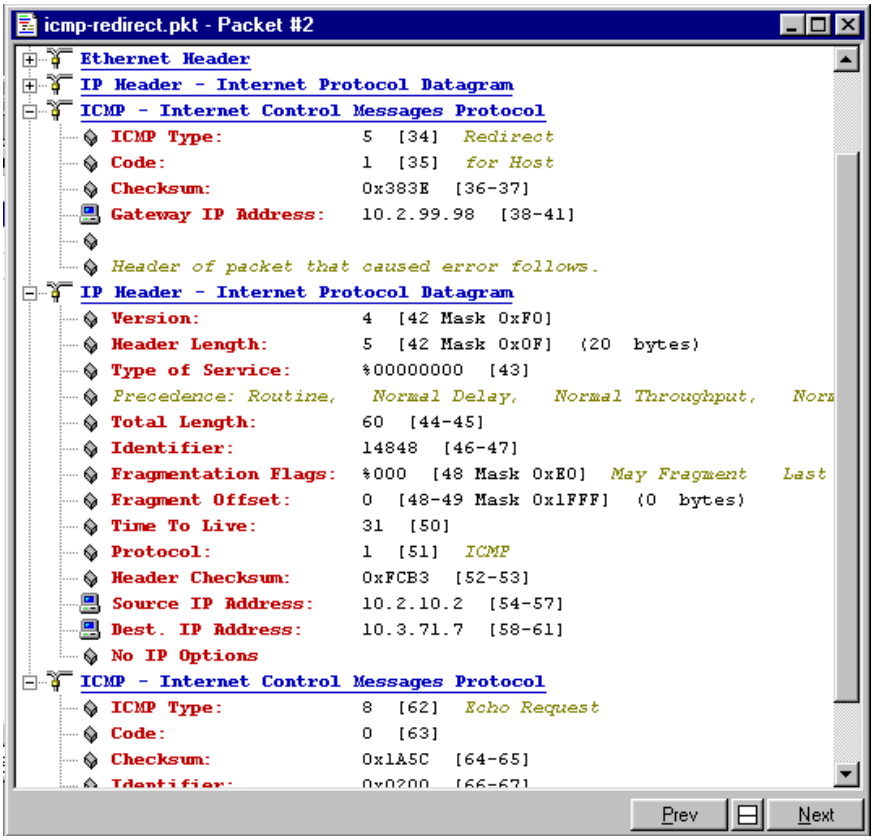

icmp-redirect.pkt - Packet #2
Ethernet Header
IP Header - Internet Protocol Datagram
ICMP - Internet Control Messages Protocol
ICMP Type: 5 [34] Redirect
Code: 1 [35] for Host
Checksum: 0x383E [36-37]
Gateway IP Address: 10.2.99.98 [38-41]
Header of packet that caused error follows.
IP Header - Internet Protocol Datagram
Version: 4 [42 Mask 0xF0]
Header Length: 5 [42 Mask 0x0F] (20 bytes)
Type of Service: %00000000 [43]
Precedence: Routine, Normal Delay, Normal Throughput,
Total Length: 60 [44-45]
Identifier: 14848 [46-47]
Fragmentation Flags: %000 [48 Mask 0xE0] May Fragment Last
Fragment Offset: 0 [48-49 Mask 0x1FFF] (0 bytes)
Time To Live: 31 [50]
Protocol: 1 [51] ICMP
Header Checksum: 0xFCB3 [52-53]
Source IP Address: 10.2.10.2 [54-57]
Dest. IP Address: 10.3.71.7 [58-61]
No IP Options
ICMP - Internet Control Messages Protocol
ICMP Type: 8 [62] Echo Request
Code: 0 [63]
Checksum: 0x1A5C [64-65]
Prev
Next

Figure 4-7: An ICMP Redirect packet.

This packet is shown using EtherPeek. The ICMP packet indicates that packet is a redirect pointing to a better gateway (10.2.99.99) for the client. The sender of the ICMP packet includes the original IP header and 8 bytes following it in the ICMP reply.

As you look further into this packet, you can see that the sender was 10.2.10.2. That device sent an ICMP echo request packet (ping) to the destination address 10.3.71.7.

Practice with the ICMP packets on your network. What are they telling you? Consider downloading *icmp-redirect.pkt* or *icmp-redirect.cap* from www.packet-level.com.

Addressing Problems

IP addressing-related problems are the most common problems in IP networks. Manually entered errors, DNS-DHCP conflicts, invalid host file entries, and so on…. These problems can be seen on the wire when a device's ARPs go unanswered or an ICMP reply comes back stating that the 'Destination Network Unknown' or 'Destination Network Unknown.'

These problems are actually not too tough to figure out because they will cause some inability to reach a device.

Routing Problems

Start your analyzer and filter on all TCP/IP traffic. What type of routing traffic do you have? Are you using RIP version 1, RIP version 2, or OSPF? Ideally, you should try to use only a single routing protocol.

Do you see many ICMP Redirect messages? These messages occur when traffic is sent to an inappropriate router (not the best way to go). If you do see lots of ICMP redirects messages, who do they go to? That is pointing at the device that may have a default gateway setting that doesn't match their typical communications pattern.

Do you see any messages indicating that a service is unreachable?

How many routing broadcasts are sent on the network (seen with RIP version 1)?

Service Discovery Problems

There are several ways that an IP device can 'discover' a service.

- They can identify the address in a local host file.
- They can use DNS to find the IP address of a device.
- They can use SLP (Service Location Protocol) to find the IP address of a device.
- With NetWare 5, they can use NDS over TCP/IP, as well.

Once the client has the address and they send communications to another device, do they receive ICMP Destination Unreachable messages? If so, why? (Look at the ICMP code coming back).

TCP Connection Problems

TCP uses a three-way handshake to establish a logical connection between devices:

Client:	SYN > to server
Server:	SYN ACK > to client
Client:	ACK

Where SYN stands for 'synchronize sequence numbers' and ACK is an acknowledgment. This process must occur successfully in order to establish a TCP connection.

TCP uses windowing (adjustable data flow sizes based on the current network congestion and the receiver capability). The Window size field in the TCP header is the receiver's buffer space available. Sometimes, you might find your network experiencing frozen windows because a client application is not freeing up buffer space fast enough.

Also… check to see that a TCP-based application (such as FTP) closes the TCP connection when you close the application. Make a nice trace of a simple FTP operation to get a feel for how TCP works.

IP Fragmentation Problems

What if you sent a 4096 byte packet from your Token Ring network to the server on an Ethernet network? Well…. unfortunately, Ethernet only supports 1518 byte packets. Imagine trying to get an elephant through a little garden hose.

Fortunately, IP has the capability of fragmenting packets (in fact, you'll see the Fragment bit in the IP header). Unfortunately, applications can set this bit to not allow fragmentation. Fortunately (again), an ICMP message can be returned indicating that 'fragmentation was needed, but the Don't Fragment bit was set.'

You can play around with this using the ping utility with the -f parameter as shown below.

Usage: **ping** [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
[-r count] [-s count] [[-j host-list] | [-k host-list]]
[-w timeout] destination-list

Options:

-t	Ping the specified host until interrupted.
-a	Resolve addresses to hostnames.
-n count	Number of echo requests to send.
-l size	Send buffer size.
-f	Set Don't Fragment flag in packet.
-i TTL	Time To Live.
-v TOS	Type Of Service.
-r count	Record route for count hops.
-s count	Timestamp for count hops.
-j host-list	Loose source route along host-list.
-k host-list	Strict source route along host-list.
-w timeout	Timeout in milliseconds to wait for each reply.

Be careful -- fragmentation is not always so hot… if a packet gets fragmented into three pieces, all three pieces must arrive at the destination before the Time to Live value in the first received fragment expires.

DHCP Problems

DHCP clients use the following basic startup routine:

1. DHCP Discover
2. DHCP Offer
3. DHCP Request
4. DHCP Ack

Look at your DHCP communications to identify that pattern. What if the discovery process fails? What if more than one address is offered (more than one DHCP server)? Did the client NACK (negative acknowledgement to not accept the address) the second address? What did the client ask for in the Request? Did the server reply with all required information?

I do love DHCP, but there are times when I think it's just plain dumb. For example, DHCP includes a 'Release' function that is used when a client is logging off - the client sends an address 'release' to the DHCP server that assigned it that address. This should free up the

address for someone else to use. What if the DHCP Release gets literally 'smashed to bits' along the way? There is not ACK required for those packets, so the DHCP client just assumes that if it sent the packet, it got there. Duh… The DHCP server won't release this address until the lease time has expired.

You can get the list of DHCP options from the reference section of www.packet-level.com*. Also, checkout the March 1999 BrainShare Daily article on DHCP. (See Appendix C.) -Laura*

Denial of Service Attacks

Denial of Service (DoS) attacks are far too common these days. A denial of service attack is based on the theory that if you hog up all the CPU of a device, that device must deny services to anyone else.

In presentations, I often let folks know that it looks like good ol' DOS is back… and it's pissed. <grin> -Laura

There are two primary types of DoS attacks:

- Brut force attacks
- Stealthy attacks

Brut force attacks are typically easy to spot with an analyzer. We'd think of a brut force attack if a device were sending out 10,000 pings in rapid succession to another device.

Stealthy attacks typically only require one or two packets to kill the target. For example, a TCP SYN packet that uses the target's IP address in both the source and destination address field of the IP header can kill some implementations of IP.

Take a look at the TCP/IP troubleshooting tools that are included with most implementations of TCP/IP. See also "TCP/IP Troubleshooting Tools" (January 1999, pp. 20-28) at www.nwconnection.com*. That article covers ping, traceroute, route, and more. -- Laura*

IPX/SPX Problems

In this section, we'll look at NetWare's IPX/SPX protocol communications and problems. First, we'll take a quick look at how IPX/SPX works and then we will go through the most common errors that you should look for on your IPX/SPX network.

NetWare IPX/SPX 101

IPX/SPX (Internetwork Packet Exchange/Sequenced Packet Exchange) was the default protocol for NetWare networks up until NetWare 5 was released. There are still millions of computers using this transport mechanism. Figure 4-8 shows the basic elements of the IPX/SPX protocol stack.

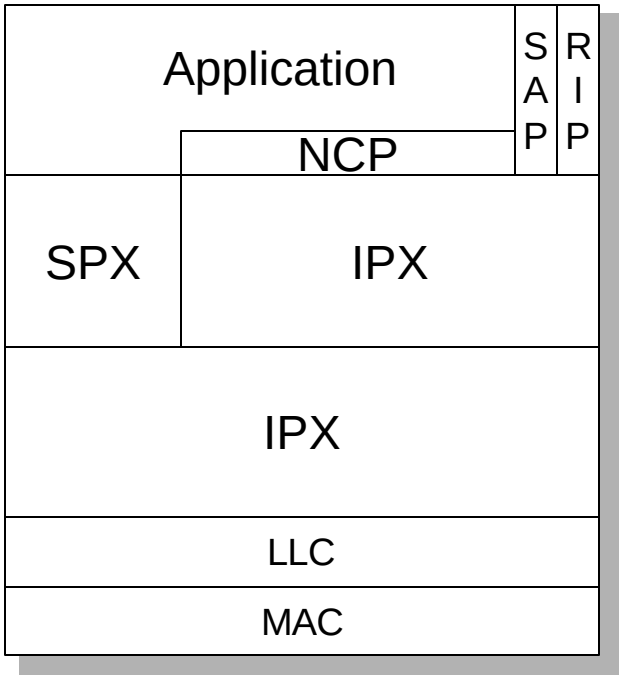


Figure 4-8: The NetWare IPX/SPX protocol elements.

The following list gives a basic overview of how NetWare clients and servers use the IPX/SPX protocol (and its related elements) to communicate.

1. Clients send a SAP (Service Advertising Protocol) broadcast to locate a NetWare 3.x server (service type 0x0004) or a Directory Server (0x0278). These SAP broadcasts are not forwarded across routers. SAP packets are never routed.

2. Only local servers respond with their internal IPX network address. If there are no local servers, a router may respond with the internal IPX network address of a server that is listed in the router's Server Information Tables. (Routers and servers build these tables through periodic SAP announcements.)

3. Upon receipt of a SAP reply, the client sends a RIP (Routing Information Protocol) broadcast to find out what route is available to get to the server's internal IPX network. Only the device with the lowest cost (based on router hops) path will respond. RIP packets are never routed.

4. NetWare client software converts application calls to NCP (NetWare Core Protocol) requests to be sent to the server. The server sends NCP replies to the client.

5. Commands typically use single request/single reply patterns. File transfer, however, can use IPX burst mode technology. Burst mode allows multiple reply packets in response to a single request packet for faster data transfer.

6. Some applications use the connection-oriented functionality of SPX to give them a packet acknowledgment system.

7. Servers periodically communicate amongst themselves to exchange NDS (NetWare Directory Services) and time synchronization information.

8. Network route and service information is periodically exchanged using SAP and either RIP or NLSP (NetWare Link Services Protocol). IPX RIP is a distance vector protocol similar to IP RIP. NLSP is a link state routing protocol similar to OSPF (Open Shortest Path First).

If you want much more detail on the entire NetWare IPX/SPX protocol stack, grab a copy of "Novell's Guide to LAN/WAN Analysis: IPX/SPX" that I wrote in 1998. The book is available through www.packet-level.com*. -Laura*

In the next section, we'll look at some of the more common NetWare problems and their possible solutions.

Excessive SAP/RIP Broadcasts

Since every device must buffer and process packets sent to the broadcast address, broadcasts affect all devices on your network. Unfortunately, Novell's SAP and RIP protocols are very 'chatty' requiring NetWare servers and IPX routers to send out broadcast updates every 60 seconds for each frame type bound to IPX.

In some cases, the client is causing the excessive SAP or RIP broadcasts (consider the case against auto frame typing later in this section).

One good exercise for an analyst is to create a capture filter on broadcast traffic and let the analyzer run for a couple of days. You should get a nice trend screen comparing all traffic to broadcast traffic. If your broadcast traffic accounts for 10% or more of total traffic during peak times, try to reduce your broadcast traffic. -Laura

---

NetWare 4.x and 5.x have configurable SAP and RIP update parameters. You can change the frequency of the update process or the packet sizes used. (By default, NetWare still uses "itty-bitty stinkin' packets" for SAP and RIP traffic.)

Routing Problems

Check your routing traffic to document how it works. Do you use IPX RIP or NLSP? Do you use both? If so, try to trim down to just one routing protocol (NLSP is preferred).

See also "NetWare Link Services Protocol: Interoperating With RIP and SAP" (Nov. 1997, pp. 36-39), "NetWare Link Services Protocol: Updating the Link State Database" (Oct. 1997, pp. 34-39), and "NetWare Link Services Protocol: Building a Link State Database" (Sept. 1997, pp. 38-45) at www.nwconnection.com*. -Laura*

How many hops away do most packets come from (look in the IPX header)? Are there any routing loops on your network? Do users access servers directly, or do they need to go through a router unnecessarily?

Service Discovery Problems

Although services can be discovered using NDS, many applications and processes still use SAP.

Look for unanswered SAP broadcasts that occurred during the bootup process. What is the cause of that ugly junk? Are there any unanswered SAP queries on your network?

SPX Problems

I really am not very fond of SPX. It's a slow, doggy connection-oriented protocol with limited functionality. Capture all your traffic and filter on SPX traffic (packet Type 5 in the IPX header). What applications appear to be using SPX?

SPX uses a two-way handshake on initialization. Capture and examine the sequence. Are there any uncompleted handshakes? (They are identified by the destination sequence number 65535.)

Are your SPX communications using minimum packet sizes and a request/reply pattern? If so, there's a second version of SPX (SPX 2) that supports a multi-packet response to increase performance for SPX applications. Check with your vendor on this.

I typically describe SPX 2 as a dog protocol and SPX 1 as a dead dog protocol. <grin> -Laura

Auto Framing Problems

I truly detest auto frame typing! Here are the reasons:

1. **Blah, blah, blah … Broadcasts**

   In order to auto detects the frame type, Novell's Client 32 and Microsoft's Client for NetWare Networks spew broadcast packets onto the network. Who processes broadcast packets? Everyone! Everyone must at least look at the packet to see if it is addressed to a functioning protocol stack within the receiving device. If the receiving device does not understand the incoming protocol (NetWare's Service Advertising Protocol, SAP, for example), the station discards the packet -phewie….). Since broadcasts go everywhere on a hubbed or switched network (provided you do not have any layer 3 functionality at that switch), there can be quite a filthy mess spewed onto the network during the morning bootup sequence.

2. **Passive self-configuring devices**

   Some products (typically network connected printers and print servers) can self-configure themselves based on the frame type they see on the network when they boot. Imagine that you have a new print server that you bring online one fine morning. Out of the 8,000 packets that are bombarding through the network, 75% of them are not the correct frame type.

3. **Translational switches (i.e. FDDI to Ethernet)**

   In the case of translational switches (switches that translate frame types from one media type to another), such as an Ethernet to FDDI switch, auto frame typing can cause some very strange connectivity problems. It may appear that everyone on the FDDI side of the switch can talk with a server, but some or all of the clients on the Ethernet side cannot talk to the server, even through the server is up and functional. It appears as if the switch is 'dropping' packets from the server to the Ethernet link. In fact, however, the switch is forwarding the communications, but not in a very smart fashion.

Ugly Frame Types

Many networks still run just fine on the ol' Ethernet_802.3 frame type. For some networks that's all they need (bless 'em for holding out so long). Others have just been avoiding the dreaded "Frame Type Migration Process" because they are either too large to make it a simple process or they just have other things to do with their weekends. Still others are holding out to move over to the Ethernet_II frame type when they change over to the TCP/IP stack as their primary transport.

There are a few pieces of bad news here for you Ethernet_802.3 frame type folks. Mainly:

- The Ethernet_802.3 frame type does not support IPX checksumming.
- The Ethernet_802.3 frame type does not support multiple protocols (yes… it's proprietary)
- The Ethernet_802.3 frame type may not be supported by third parties forever.

See the Greypaper entitled, "Why You Should Avoid the Ethernet_802.3 Frame Type" at www.packet-level.com *for more information. -Laura*

Spend some time looking at your NetWare traffic to see if you use the Ethernet_802.3 frame type only.

Remember, that the ultimate reference book on IPX/SPX communications really is "Novell's Guide to LAN/WAN Analysis: IPX/SPX" - I can say that because there aren't many folks writing on the IPX/SPX analysis stuff these days. This has caused some folks to question whether IPX/SPX is a dying protocol.

Trace Files to Practice On

We have a number of trace files at www.packet-level.com.

I suggest that you download and review these files. Keep checking out there for more files as we release our lab files (with some documentation) publicly.

When you view these trace files, remember to focus on the summary portion first - get a general feel for the traffic and look for recognizable patterns. Then begin to focus on one conversation. Make sure you get a good screen capture utility that will help you document your findings in a graphical manner, as well.

If you have a suggestion of a trace that you'd like to see, send it to feedback@packet-level.com. -Laura

Chapter 4 Test

Here we go again. This test covers all four of the topics in this chapter - Ethernet/802.3, Token Ring/802.5, IPX/SPX and TCP/IP. Do your best.

The answers are in Appendix A - but don't cheat - really give it a good try before looking up the answers.

4.1 You are analyzing an Ethernet network that is experiencing a high number of collisions, but relatively low utilization peaks. What could be the problem?

_____

_____

_____

4.2 Where can you get the CSMA/CD specification?

_____

4.3 What is the difference between an Ethernet collision fragment and a short packet?

_____

_____

_____

4.4 You are analyzing a Token Ring network that is experiencing a high number of line errors reported from the server's address. How can you fix this problem?

_____

_____

_____

4.5 You have captured a soft error report MAC frame that reports a token error. Based on this packet, what do you know about the reporting station?

_____

_____

_____

4.6 You see hundreds of SAP broadcasts on your network. Inside the IPX header, you notice that the source network address is 0x00-00-00-00 on these packets. What is most likely the purpose of these packets? How do you eliminate them from the network?

_____

_____

_____

4.7 What is the most wonderful protocol of the entire TCP/IP protocol suite (speaking from an analyst's perspective, of course)?

_____

_____

_____

4.8 Define the difference between UDP and TCP.

_____

_____

_____

4.9 Define the three packets seen in the TCP handshake.

_____

_____

_____

4.10 How can you perform a test to see if IP fragmentation is required to communicate between your client on a Token Ring network and a server on a remote Token Ring network?

_____

_____

_____

CHAPTER 5

Analysis Tools and Exercises

This chapter contains reference to numerous products that are available online and as part of “Laura’s Lab Kit.”

If you have received this book in either the hardcopy format or as part of a multi-part CD set, you have received the demo software referenced in this chapter.

If you downloaded this book, you can get the demo software directly from the vendors and their websites -- the sites are listed in the following pages. Numerous trace files are zipped up with the book however.

The next page lists the software included in Laura’s Lab Kit.

Laura's Lab Kit

The following applications are included with Laura's Lab Kit although not all the applications are covered in the exercises listed in this chapter.

- EtherPeek Demo (see EPWlicense.txt) [www.wildpackets.com]
- TokenPeek Demo (see TPWlicense.txt) [www.wildpackets.com]
- NeoTracePro Demo [www.neoworx.com]
- NetScanTools Demo [www.nwpsw.com]
- PacketScrubber Demo [www.wildpackets.com]
- ProConvert Demo [www.wildpackets.com]
- Sam Spade [www.samspade.org/ssw/]
- IPCalc for Windows [www.wildpackets.com]
- IPCalc Demo for Palm [www.wildpackets.com]
- NetDoppler Demo [www.wildpackets.com]
- NetSense Demo [www.wildpackets.com]
- iNetTools Demo [www.wildpackets.com]
- WebStats Demo (see WSlicense.txt) [www.wildpackets.com]
- WatchPoint Lite Demo [www.wildpackets.com]

Wow! Sure looks like I use lots of WildPackets products, eh? Well... I do. I would have also included a Sniffer Basic demo, but at the time I revised this book, they decided not to release a demo version to the public. Maybe that will change in the future.

Analysis Exercises

This chapter contains lab exercises for many of the products listed earlier. Follow the step-by-step instructions to get a good feel for the products and how to use them for troubleshooting and optimizing your network.

The following lists the lab tests that you will find in this document.

Exercise 1: Etherpeek	Capturing Packets
	Checking Out Your Protocols and Packet Sizes
	Opening Saved Trace Files
Exercise 2: NeoTracePro	Checking out the Path to Pod-books.com
Exercise 3: NetScanTools	Using WhoIs and Learning to Gather Information
Exercise 4: PacketScrubber	Scrubbing IP Addresses from an HTTP Trace File
Exercise 5: ProConvert	Converting an HP Advisor Trace File
Exercise 6: Sam Spade	Explore Domain Information for laurachappell.com
Exercise 8: SubnetCalculator	Defining Subnets for 191.15.0.0.

Exercise 1: EtherPeek

EtherPeek is built and distributed by WildPackets, Inc. (formerly the AG Group). Easy-to-use and full of chart and graphing capabilities, EtherPeek is the up-and-coming hot protocol analysis tool.

It should be noted that EtherPeek has been around for a long time - heck, we used to use it in the IS lab when I was at Novell - and you know that was ages ago. -- Laura

Naturally, being a packet-lovin' gal, I want you to immediately capture some packets. Figure 5-1 shows the results of my test (in which I captured my own web-browsing traffic).

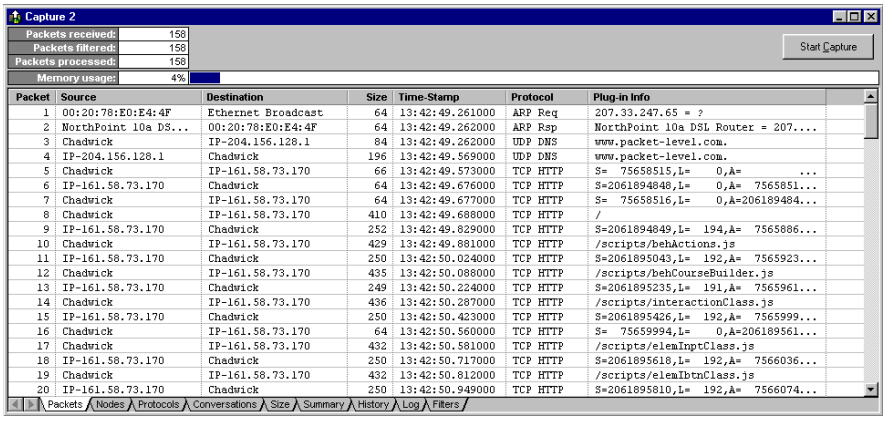

Capture 2
Packets received: 158
Packets filtered: 158
Packets processed: 158
Memory usage: 4%
Start Capture
Packet | Source | Destination | Size | Time-Stamp | Protocol | Plug-in Info
1 | 00:20:78:E0:E4:4F | Ethernet Broadcast | 64 | 13:42:49.261000 | ARP Req | 207.33.247.65 = ?
2 | NorthPoint 10a DS... | 00:20:78:E0:E4:4F | 64 | 13:42:49.262000 | ARP Rsp | NorthPoint 10a DSL Router = 207....
3 | Chadwick | IP-204.156.128.1 | 84 | 13:42:49.262000 | UDP DNS | www.packet-level.com.
4 | IP-204.156.128.1 | Chadwick | 196 | 13:42:49.569000 | UDP DNS | www.packet-level.com.
5 | Chadwick | IP-161.58.73.170 | 66 | 13:42:49.573000 | TCP HTTP | S= 75658515,L= 0,A= ...
6 | IP-161.58.73.170 | Chadwick | 64 | 13:42:49.676000 | TCP HTTP | S=2061894848,L= 0,A= 7565851...
7 | Chadwick | IP-161.58.73.170 | 64 | 13:42:49.677000 | TCP HTTP | S= 75658516,L= 0,A=206189484...
8 | Chadwick | IP-161.58.73.170 | 410 | 13:42:49.688000 | TCP HTTP | /
9 | IP-161.58.73.170 | Chadwick | 252 | 13:42:49.829000 | TCP HTTP | S=2061894849,L= 194,A= 7565886...
10 | Chadwick | IP-161.58.73.170 | 429 | 13:42:49.881000 | TCP HTTP | /scripts/behActions.js
11 | IP-161.58.73.170 | Chadwick | 250 | 13:42:50.024000 | TCP HTTP | S=2061895043,L= 192,A= 7565923...
12 | Chadwick | IP-161.58.73.170 | 435 | 13:42:50.088000 | TCP HTTP | /scripts/behCourseBuilder.js
13 | IP-161.58.73.170 | Chadwick | 249 | 13:42:50.224000 | TCP HTTP | S=2061895235,L= 191,A= 7565961...
14 | Chadwick | IP-161.58.73.170 | 436 | 13:42:50.287000 | TCP HTTP | /scripts/interactionClass.js
15 | IP-161.58.73.170 | Chadwick | 250 | 13:42:50.423000 | TCP HTTP | S=2061895426,L= 192,A= 7565999...
16 | Chadwick | IP-161.58.73.170 | 64 | 13:42:50.560000 | TCP HTTP | S= 75659994,L= 0,A=206189561...
17 | Chadwick | IP-161.58.73.170 | 432 | 13:42:50.581000 | TCP HTTP | /scripts/elemInptClass.js
18 | IP-161.58.73.170 | Chadwick | 250 | 13:42:50.717000 | TCP HTTP | S=2061895618,L= 192,A= 7566036...
19 | Chadwick | IP-161.58.73.170 | 432 | 13:42:50.812000 | TCP HTTP | /scripts/elemIbtnClass.js
20 | IP-161.58.73.170 | Chadwick | 250 | 13:42:50.949000 | TCP HTTP | S=2061895810,L= 192,A= 7566074...
Packets Nodes Protocols Conversations Size Summary History Log Filters

Figure 5-1. The captured packets.

The Capture Window is pretty standard stuff on EtherPeek - source, destination, timestamp (absolute), protocol and summary information. The tabs at the bottom of the capture window are really important - they give you serious in-depth information into the top talkers, protocols, devices, packet size, communication summaries, histories, error/alarm logs and filters related to the captured data.

Capturing Packets

Step 1: **Launch** the EtherPeek Demo program. Click **OK** to accept the demo limitations shown.

Step 2: Click on the **Packet Capture button** . Click **OK** to accept the Capture Buffer options.

Step 3: Click on **each of the tabs** to see the wealth of information available.

Let me give you a heads-up here... EtherPeek can provide you with a tremendous number of statistics - in many cases, you can get a graphical representation of those statistics - perfect for that report to the boss. -- Laura

---

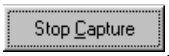
Step 4: Click the **Packets Tab**

. This enables you to see the packets as they are being captured into the buffer.

Step 5: Click the **Start Capture button**

. You're now capturing all the traffic that your network interface card sees. Capture at least 100 packets. (The demo is limited to 250 packets anyway.)

For information on supported hardware, see www.wildpackets.com/support/hardware/etherpeek_win. *-- Laura*

Step 6: Click the **Stop Capture button** 

. Wow! Look at all that cool data! Scroll through to see what you got… kinda like fishing, eh? Next we'll check out the protocols seen on your network. Try this step while you toggle to another window and do some web browsing or maybe download your email.

Step 7: Click the **Close window button** to close the Capture window.

Checking out your Protocols and Packet Sizes

It's easy to find out what protocols are running on the network. The Protocol Statistics window offers protocol details based on protocol type (in a nice layered design that shows the encapsulation of protocols), percentage, bytes and packets.

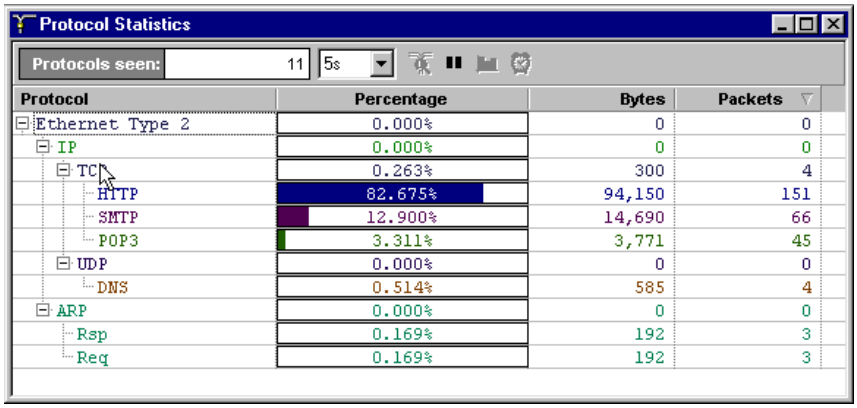


Figure 5-2. Checking out the Protocol Statistics

Step 1: Click on the **Protocol Statistics button** .

You can also click on the Protocols tab in your trace buffer window to see exactly which protocols were running on your network during your packet capture operation. -- Laura

Step 2: Click on the **pie chart icon** on the menubar to view packet size distribution statistics, as shown in Figure 5-3. As you probably know, a bunch of little itty-bitty stinkin' packets is not a good thing - especially if there are a lot of file transfers going on.

The pie chart is simple to understand.

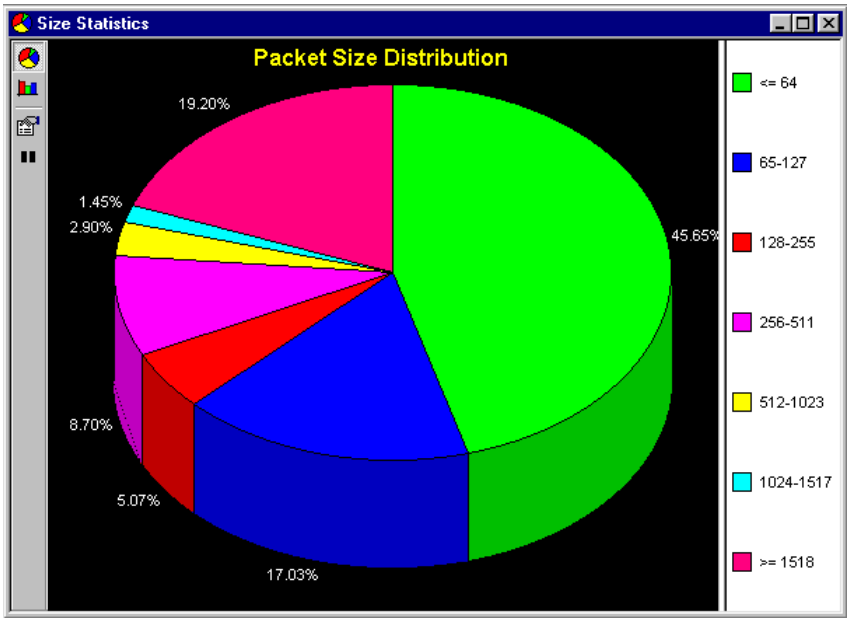


Figure 5-3. Checking out the Protocol Error Statistics

Step 3: Click on the **Options button** to configure the pie chart differently.

Opening Saved Trace Files

This book includes some packet traces in a \tracefiles directory. There are several interesting traces including the following:

arp.pkt	IP address test and standard ARP operation.
dhcpboot.pkt	Standard DHCP boot sequence
dhcprenewtorebind.pkt	DHCP client cannot renew or rebind his address.
dns1.pkt	Standard DNS lookups
fin-4pkts.pkt	TCP connection close (4-packet sequence)
fragments.pkt	Fragmented ICMP echo requests
http.pkt	HTTP communications
icmplab.pkt	ICMP Destination Unreachable response
portscan.cap	TCP port scan (Sniffer format)
tcp-backoff.pkt	TCP backoff algorithm
tcp-handshake.pkt	TCP 3-way handshake process
telnet1.pkt	Telnet sequence

In this next exercise, you open up *icmplab.pkt* to view an ICMP message. (Remember, these trace files are also available online at www.packet-level.com.)

Step 1: Click on the File Open button .

Step 2: Double-click on ***icmplab.pkt*** in the tracefiles directory. There are only two packets in this trace file, as shown in Figure 5-4.

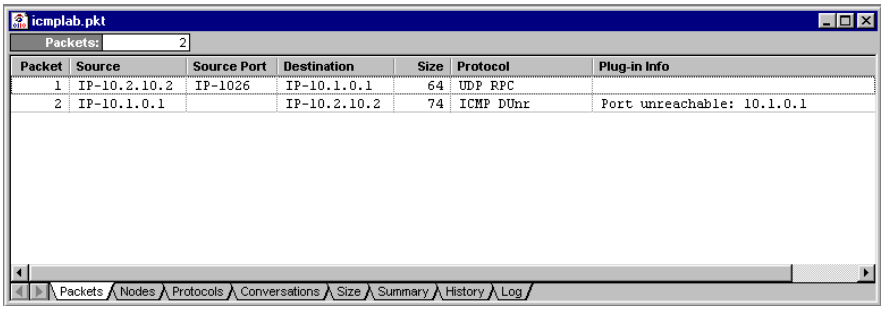

icmplab.pkt
Packets: 2
Packet Source Source Port Destination Size Protocol Plug-in Info
1 IP-10.2.10.2 IP-1026 IP-10.1.0.1 64 UDP RPC
2 IP-10.1.0.1 IP-10.2.10.2 74 ICMP DUnr Port unreachable: 10.1.0.1
Packets Nodes Protocols Conversations Size Summary History Log

Figure 5-4: The summary information is shown when you open a trace file.

Step 3: Double-click **anywhere** along the line of the first packet to open up the packet decode. **Scroll** through the packet to find the UDP destination port field.

What is the destination port number? 111 - Sun Remote Procedure Call. Next, we'll look at the ICMP response.

Step 4: Click on the **Next button**

at the bottom of the decode window.

Now you can see the ICMP response. This response contains the original request's IP header and UDP header.

The reason why this Sun RPC request was not honored is obvious - the responder is not listening on port 111 (Destination Unreachable; Port Unreachable).

Take a look through some of the other trace files to get a taste for the other communications.

For more EtherPeek exercises, download the "10 Cool Things You Can Do With the EtherPeek Demo" online in the archives section at www.packet-level.com*. -- Laura*

This is only a quick view of EtherPeek. There are so many things you can do with this product! I highly recommend it.

For more information on EtherPeek, visit www.wildpackets.com.

Exercise 2: NeoTrace Pro

This is a cool product. Very well written and graphically superior to the other tracing programs I've used.

This exercise is based the trial version of NeoTrace Professional. It is fully functional with no disabled or altered features, but will expire 30 days after it is installed. -- Laura

During the installation, you'll notice that NeoTrace Pro wants to know where you are located - give it your country and postal code and you are set! If you are a true geek, you can switch to advanced mode and set your location by longitude and latitude, as you can see in Figure 5-5.

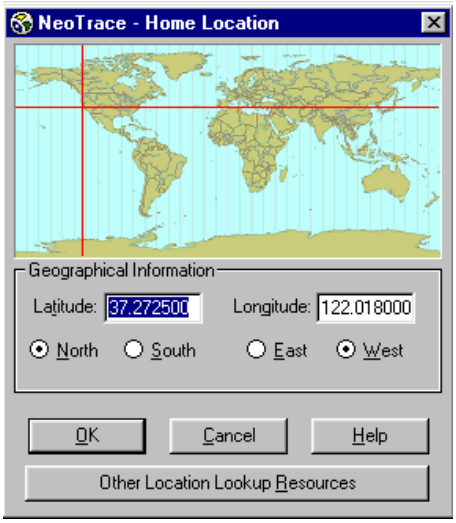


Figure 5-5: Be a geek - use your latitude and longitude to set up NeoTrace Pro!

Checking out the Path to Podbooks.com

Once you've set your NeoTrace location, the main NeoTrace window appears Now you can begin tracing the path to any location. In Figure 5-6, we've traced the path from the to podbooks.com, my publishing company. Interestingly, you can see that we have a redirect set up for podbooks.com - the actual site that handles podbooks.com is imagitech.stores.yahoo.com (just a temporary location for the store).

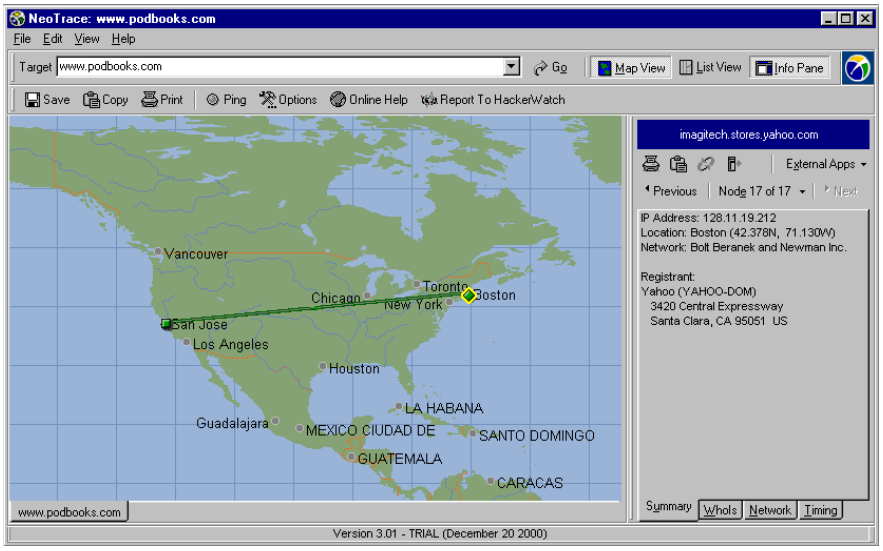


Figure 5-6. Tracing your way to podbooks.com.

Now you try performing a trace.

Step 1: **Launch** the NeoTrace Pro demo program.

Step 2: Type **www.podbooks.com** in the Target field.

Click **Go**

.

While the trace is going on, the map will adjust to show the path all the way to the destination. The summary information includes the IP address and registrant of the destination.

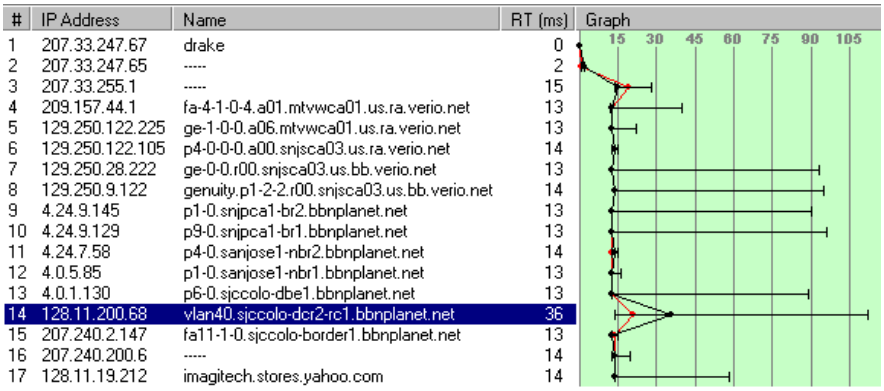
Step 3: Click on the **Whois tab**

to list the registrant of this domain name.

Step 4: Click on the **Previous** and **Next** buttons to follow the path back from the destination to your source.

Step 5: Click the **List View button**

to view the response times to each of the routers along the path, as shown in Figure 5-7.



#	IP Address	Name	RT (ms)
1	207.33.247.67	drake	0
2	207.33.247.65	-----	2
3	207.33.255.1	-----	15
4	209.157.44.1	fa-4-1-0-4.a01.mtvwca01.us.ra.verio.net	13
5	129.250.122.225	ge-1-0-0.a06.mtvwca01.us.ra.verio.net	13
6	129.250.122.105	p4-0-0-0.a00.snjsca03.us.ra.verio.net	14
7	129.250.28.222	ge-0-0.r00.snjsca03.us.bb.verio.net	13
8	129.250.9.122	genuity.p1-2-2.r00.snjsca03.us.bb.verio.net	14
9	4.24.9.145	p1-0.snjpca1-br2.bbnplanet.net	13
10	4.24.9.129	p9-0.snjpca1-br1.bbnplanet.net	13
11	4.24.7.58	p4-0.sanjose1-nbr2.bbnplanet.net	14
12	4.0.5.85	p1-0.sanjose1-nbr1.bbnplanet.net	13
13	4.0.1.130	p6-0.sjccolo-dbe1.bbnplanet.net	13
14	128.11.200.68	vlan40.sjccolo-dcr2-rc1.bbnplanet.net	36
15	207.240.2.147	fa11-1-0.sjccolo-border1.bbnplanet.net	13
16	207.240.200.6	-----	14
17	128.11.19.212	imagitech.stores.yahoo.com	14

Figure 5-7: Examining the roundtrip times to the routers along the path.

Not all destination systems respond as expected to the NeoTrace process. For example, try tracing the path to www.microsoft.com. *Then click on ping to try a simple ping to* www.microsoft.com. *As you'll see, they don't respond to pings. Most likely, they've been bombarded by a few too many ping-based attacks. -- Laura*

Try watching the List View while you trace the path to some of the web sites you visit.

For more information or to order NeoTrace Pro, visit www.neoworx.com.

Exercise 3: NetScanTool Pro

There are two versions of NetScanTools - NetScanTools Standard Edition and NetScanTools Pro. This exercise is based on the NetScanTools Standard Edition.

The feature comparison chart at http://www.netscantools.com/nstcomparison.html provides details on the difference between the two versions. Trust me, the Pro version is well worth the extra money.

I highly recommend this product. It's one of the tools that I regularly reference to check connectivity, test open ports and resolve DNS information. -- Laura

Using WhoIs and Learning to Gather Information

In this lab exercise, you'll learn how to use NetScanTools to train yourself on how to use NetScanTools - pretty nice. In fact, the Help files written for NetScanTools is by far the best and most technically useful Help files I've ever seen.

Figure 5-8 shows the NetScanTools Help Wizard that you'll use to learn the product.

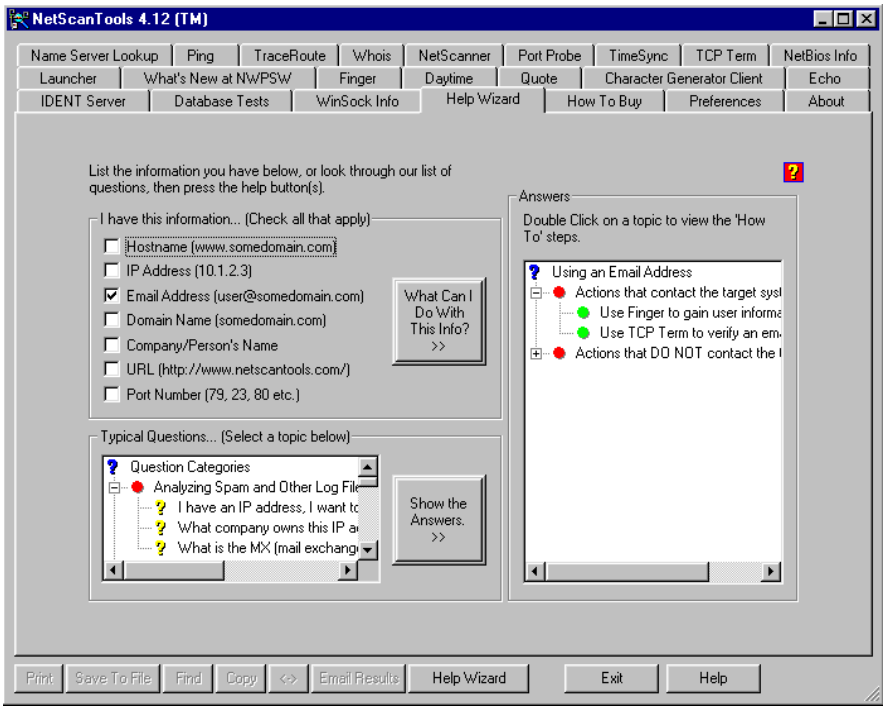


Figure 5-8: Run the Help Wizard to learn to use the product.

Step 1: **Launch** the NetScanTools demo program.

Step 2: Click on the **Help Wizard tab**

.

Step 3: On the left side of the wizard screen there are two sections:

- I have this information… (Check all that apply)
- Typical Questions… (Select a topic below)

In the top box, check the box next to **Domain Name** (somedomain.com).

Step 4: Click "**What Can I Do With This Info? >>**"

The window on the right lists two sets of options:

- · Actions that contact the target system.
- · Actions that DO NOT contact the target system.

Click the **+** next to the second option - **Actions that DO NOT contact the target system**.

There are four options listed. Three processes use the Name Server lookup tool and one process lists the WhoIs tool.

We'll use the WhoIs tool to see who registered a domain name.

Step 5: Click the **WhoIs tab**.

Step 6: Enter **laurachappell.com** in the Query field. Click the **Query button**.

The results returned include the domain registrar, the registrant, the date the record was updated, etc. Figure 5-9 shows the results screen.

The 'Smart WhoIs' mode uses the domain extension to determine which WhoIs Server should be sent the query. If you are trying to query on nwpsw.com, for example, it sends the query to whois.networksolutions.com. If you are trying to query on af.mil, it sends the query to whois.nic.mil and so forth. -- Laura

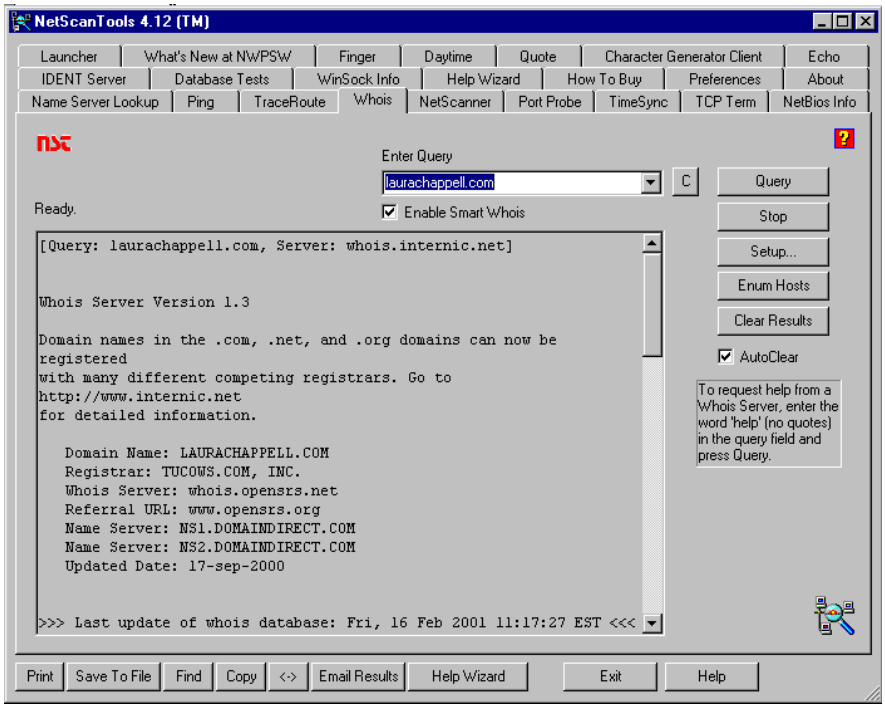


Figure 5-9: WhoIs

Return to the Help Wizard and check out all the functions you can perform using this wonderful tool.

For the price (1-10 Quantity Unit Price is $150), this purchase is a no-brainer. Get this product today!

For more information or to order NetScanTools or NetScanTools Pro, visit www.nwpsw.com.

Exercise 4: PacketScrubber

When you're going to send that trace file to someone to analyze or show it in a presentation, I highly recommend that you sanitize the file - scrub the file - remove your company's IP addresses and any other 'private' information from the trace file.

PacketScrubber was written by Scott Haughdahl. Scott is the founder of Net3Group, which was recently acquired by WildPackets. Scott wrote an excellent book entitled, "Network Analysis and Troubleshooting." The ISBN number is 0201433192 - consider adding this title to your library! -- Laura

Figure 5-10 shows the main window of PacketScrubber. Simple, eh? Just enter the trace file name and click the Scrub button. PacketScrubber automatically scrubs the IP address, by default (replacing private addresses with the public addresses defined by RFC 1918, "Address Allocation for Private Internets."

PacketScrubber also automatically renames the file, adding "_scrubbed" to the filename so you can easily tell the difference between a scrubbed and an unscrubbed file.

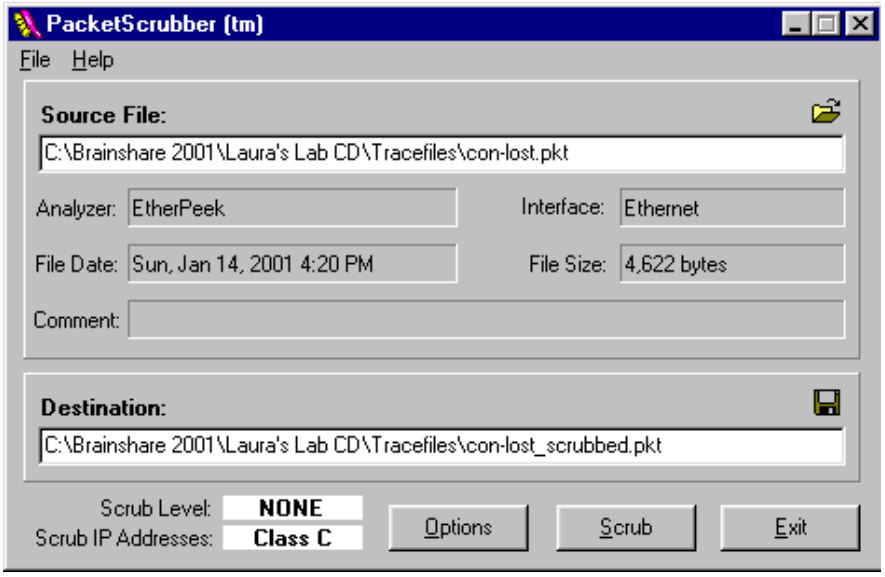


Figure 5-10: The basic PacketScrubber window.

There are numerous options that can be employed when you are running PacketScrubber. Clicking on the Options button reveals the various layers and protocols that you can scrub in a packet. Figure 5-11 shows the Options window.

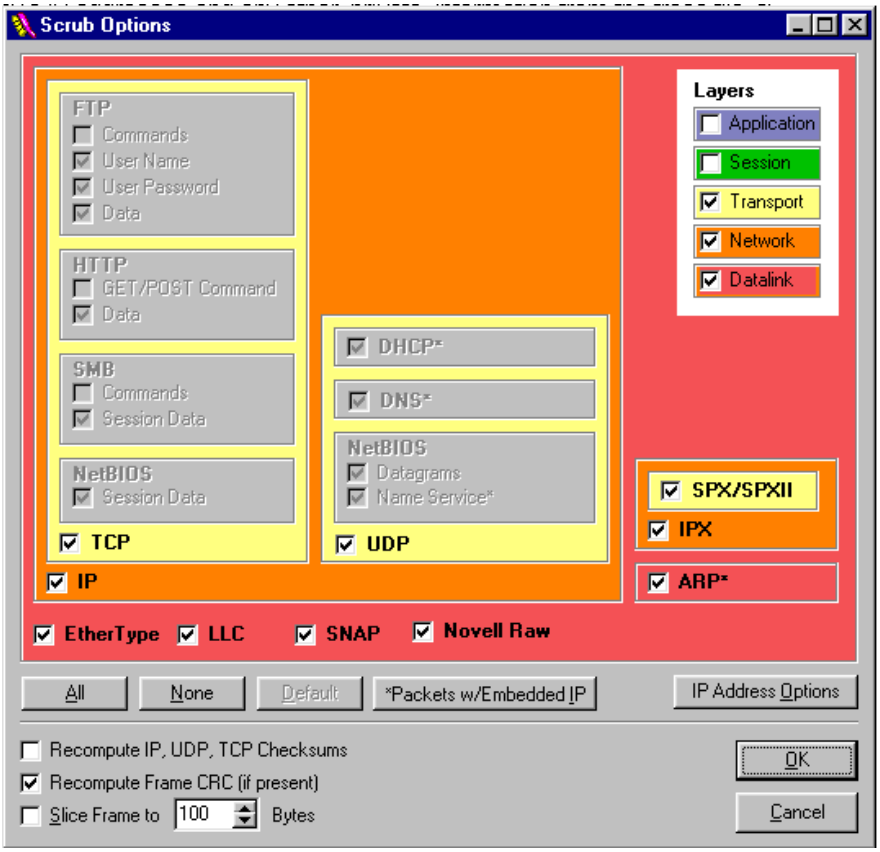


Figure 5-11: You can scrub a packet raw with PacketScrubber.

Use the options when you really want to sanitize a packet all the way to the application layer. For example, perhaps you don't want the DNS name included in the DNS query trace that you captured. Simply click the Session button in the layers box and ensure the DNS box is checked is selected.

By default, PacketScrubber cleans up through the transport layer (UDP/TCP) only.

Scrubbing IP Addresses from an HTTP Trace File

Now it's time for you to scrub a trace file using PacketScrubber. Get the file *http.pkt* from the www.packet-level.com web site (unless it was included with this book). This is an unscrubbed trace file of an HTTP session. Follow the directions in the lab exercise to scrub the file for distribution.

You may want to open the file *http.pkt* to see the unscrubbed version before you scrub it. Since you will rename the file during this scrub operation, however, you can always check out the original unscrubbed version later, if desired.

Step 1: **Launch** the PacketScrubber demo.

Step 2: Click on the **file open button**.

Step 3: **Point** to the CD drive and the tracefiles directory in the "Look in" window. Double-click on the file ***http.pkt***. Click **Open**.

This file is placed in the Source File field. The Destination file field is automatically filled out with the packet name *http_scrubbed.pkt*.

Step 4: You will need to change the destination directory for the scrubbed file. Click the **save button**. Enter the **directory** where you want to save your scrubbed trace file. Click **Save**.

Hint: Consider making a directory called "scrubbed" for all your scrubbed trace files.

Step 5: Click **Scrub**.

Step 6: Open the scrubbed file using the EtherPeek demo (select **File > Open** and **point** to the file with '_scrubbed' in the name. Note that the source IP addresses are now all changed to start with 192.168.0.

Try one on your own. The trace file *ftp.cap* in the tracefiles directory has not been scrubbed.

For more information on PacketScrubber, visit www.wildpackets.com.

Exercise 5: ProConvert

ProConvert is used to convert trace files from one manufacturer's format to another. For example, if someone sends me a trace file they obtained using Shomiti's Surveyor, I may convert it to Ether-Peek or Sniffer format so I can read it on my portable.

The ProConvert demo included on the CD allows you to convert the first 25 packets with any supported trace file. The following table lists the supported trace file formats:

Company	Analyzer	File Suffix
Agilent Technologies	Agilent LAN Analyzer	*.cap
Digitech, Inc.	LAN900™	*.dat
Fluke	Protocol Inspector®	*.cap
HP/Agilent Technologies	Internet Advisor LAN™ (Win95/98)	*.dat, *.eth, *.trn
IBM	DatagLANce™	*.dgc
Microsoft	Network Monitor (SMS 1.2 and 2.0)	*.cap
Network Associates	Sniffer®	*.enc, *.trc, *.fdc, *.cap
Network Instruments	Observer®	*.bfr
Novell	LANalyzer® for Windows	*.tr1
Precision Guesswork	LANWatch™	*.dmp
Shomiti	Surveyor™	*.cap
Sun Microsystems	snoop	*.snoop, *.cap
TTC	Fireberd500™	*.pkt
TTC	Fireberd500™ PC	*.cap
UNIX and Linux Platforms	tcpdump	*.dmp
Wavetek Wandel Goltermann	DA-30™ (DOS version)	*.an1
Wavetek Wandel Goltermann	Domino™	*.cap
Wavetek Wandel Goltermann	LinkView™ Pro	*.cap
WildPackets	EtherPeek™/TokenPeek™	*.pkt, *.tpc

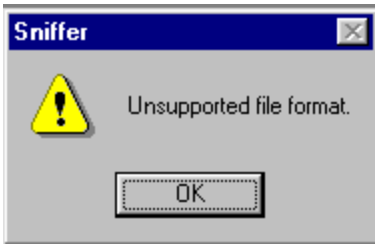
This tool is really a 'must have' for an analyst. Many of the vendors listed above have trace files on their web sites. Even if you don't have their analyzer, you can convert the trace files to the analyzer format you do support.

ProConvert was also written by Scott Haughdahl. -- Laura

Converting an HP Advisor Trace File

As you already know, the CD includes various traces in the \trace-files directory. One of the files, *DHCPJerkTakesAddressNoPermission.dat*, is in the HP Advisor .dat file format.

If you try to open this file with the Sniffer or EtherPeek demo, you receive these messages:



or

In this exercise, we want to convert this .dat trace file format to an EtherPeek .pkt file format.

Step 1: **Launch** the ProConvert demo.

Step 2: Click on the **file open button**.

Step 3: **Point** to the CD drive and the tracefiles directory in the "Look in" window. Double-click on the file ***DHCPJerkTakesAddressNoPermission.dat***. Click **Open**.

This file is placed in the Source File field. The Destination file field is automatically filled out with the same information. You'll need to change this in Step 4.

Step 4: Click the **save button**. Enter the **directory** where you want to save your .pkt trace file. Click **Save**.

Hint: Consider making a directory called "traces" for all your converted trace files.

Step 5: Click on the **Format** drop-down menu arrow. Scroll down to **click on WildPackets EPW/ TPW** (EtherPeek for Windows/TokenPeek for Windows). The Destination packet file extension automatically changes to .pkt, the EtherPeek/TokenPeek trace file extension.

Step 6: Click the **Convert button**.

If desired, you can open your converted file in EtherPeek to check your conversion.

Remember, the demo only converts the first 25 packets - you really need the full product to be ready for any conversion situation.

For more information on ProConvert, visit www.wildpackets.com.

Exercise 6: Sam Spade

Sam Spade is an integrated network query tool for Windows 95, 98, NT4.0 & Windows 2000. It's freeware.

Sam Spade includes the following tools:

ping	DNS zone transfer
nslookup	SMTP relay check
whois	Usenet cancel check
IP block whois	website download
dig	website search
traceroute	email header analysis
finger	Email blacklist query
SMTP VRFY	Abuse address query
web browser	S-Lang scripting
keep-alive	Time
nslookup	

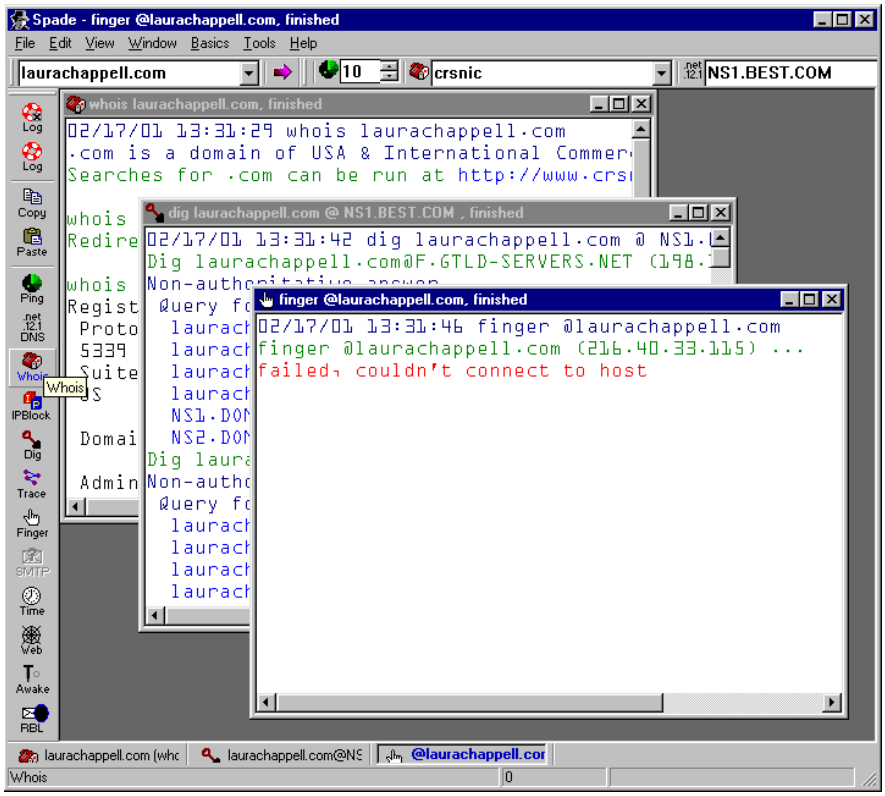


Figure 5-12: Enter the target and then click on the action, such as Dig, Whois, and Trace.

Explore Domain Information of laurachappell.com

In this exercise, we'll look at how Sam Spade can be used to check out a domain name, laurachappell.com.

Step 1: **Launch** Sam Spade.

Step 2: Enter **laurachappell.com** in the target window (in the upper left corner).

Step 3: Click the **ping button**.

By default, Sam Spade sends 10 ping packets to the destination.

Step 4: Click the **Whois button**.

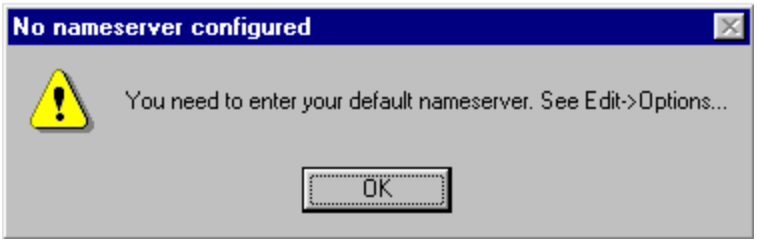
You will receive the registration information for the domain name laurachappell.com. You'll notice that I registered this name with Domain Direct.

Step 5: Click on **IP Block button**.

Sam Spade performs Domain System inverse mapping to identify the owner of the IP block associated with laurachappell.com.

Step 6: Click on the **Dig button**.

You will receive an error message indicating that the nameserver is not configured.



Click back on the **Whois Window**. The nameservers are listed at the bottom of the window. We can use either of these nameservers to get more information on laurachappell.com.

```
Domain servers in listed order:
   NS1.DOMAINDIRECT.COM    216.40.33.21
   NS2.DOMAINDIRECT.COM    216.40.33.22
```

Step 7: Click **Edit > Options**. The cursor will be in the nameserver field. Enter **ns1.domaindirect.com**.

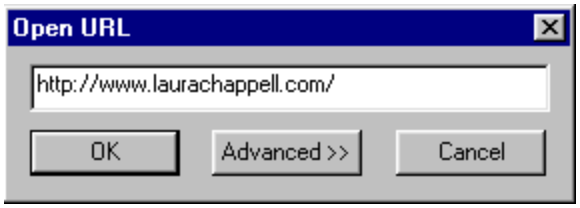
Step 8: Click on the **Dig button**

. Now you will receive lots of information regarding the laurachappell.com domain.

Step 9: Finally, we'll look at the Web view option.

Click on the **Web button**.

You are prompted to enter the desired URL. Enter **http://www.laurachappell.com**. Click OK.



This www.laurachappell.com website has a URL redirector that points to www.packet-level.com.

Play around with Sam Spade. It's a nice tool. Not as complete as NetScanTools Pro, but a good starting product.

The Sam Spade Help includes several interesting Tutorials, including Spam-Tracking 104: a Spammer Unmasked. -- Laura

For more information on Sam Spade, visit www.samspade.org/ssw/.

Exercise 7: IP SubnetCalculator

IP SubnetCalculator was another hot tool written by Scott Haughdahl - don't' ya love that guy?

Use the IP SubnetCalculator to determine how to subnet your network address for the desired number of hosts and subnets.

In our example, we are using the Class B address 191.15.0.0. Figure 5-13 shows the Subnet/Hosts listing available for 191.15.0.0.

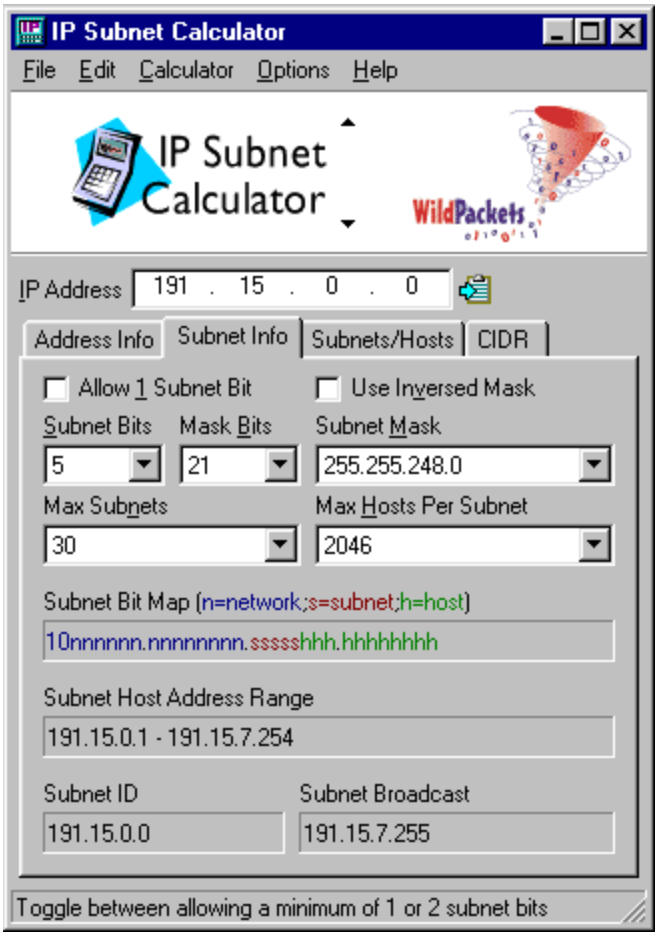


Figure 5-13: The Subnet/Hosts list.

In this exercise, we are going to figure out how to subnet 191.15.0.0 to provide for at least 24 different subnets (with a little extra just in case we grow faster than thought).

Defining subnets for 191.15.0.0

As mentioned, we're working with 191.15.0.0. We plan to have at least 24 subnets in our company internetwork. We've got to subnet out the address appropriately.

Step 1: **Launch** IP Subnet Calculator.

Step 2: Enter **191.15.0.0** in the IP address field.

Step 3: Click on the **Subnet Info tab**.

Step 4: Click the **down arrow** by the Max Subnets field as shown in Figure 5-14. Click on the value **30** to allow for the desired 24 subnets and a few extra for growth.

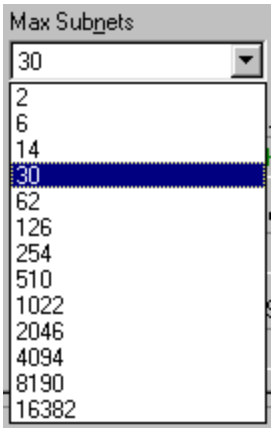


Figure 5-14: Click on the desired number of subnets.

IP Subnet Calculator automatically calculates the maximum number of hosts per subnet at 2046 - plenty for our needs.

The subnet mask we'll need to use is 255.255.248.0.

Now we want to know how to number each of the hosts in each subnet.

Step 5: Click on the **Subnet/Hosts table tab** to see the possible host addresses and subnet groups, as shown in Figure 5-15.

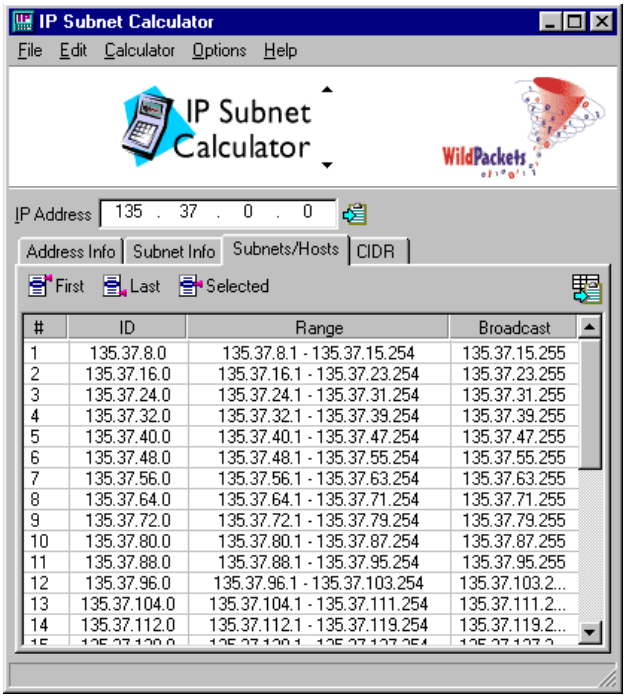


Figure 5-15: The Subnet Range field lists all the possible host addresses for each subnet.

Play around with this tool. There may be other subnet calculators out there on the Internet, but I believe Scott did a wonderful job designing and building this one. It's my favorite by far.

For more information on Subnet Calculator, visit www.wildpackets.com.

Well, that's it for this set of lab exercises. If you find any edits/ changes that need to be incorporated, please let me know (send email to lchappell@packet-level.com).

APPENDIX A

Answers to Quizzes

The appendix provides the answers to all the chapter quizzes.

Chapter One Answers

1.1 How do you reduce the number of packets that are captured into the trace buffer?

Reduce the number of packets that are captured in to the trace buffer by applying a capture filter (pre-filter) to define the traffic you are interested in. For example, if you are just looking at TCP/IP traffic, apply an IP capture filter.

1.2 What is the purpose of the display filter?

The purpose of the display filter is to highlight a select group of packets out of the trace buffer for further study. For example - let's say that you filled your trace buffer with 8 MB data. Who has time to sort through 8 MB of data? So, you apply a display filter to obtain a subset of the whole mess. Maybe you're interested in broadcast traffic only, for example. In that case, you'd filter out all traffic addressed to 0xFF-FF-FF-FF-FF-FF.

1.3 Where is the trace buffer located?

The trace buffer is in memory on the analyzer system itself. Many analyzer vendors enable you to capture to disk - in that case, the trace buffer is partially in memory and partially on disk (as the packets are being written to a file). Be careful of this option - I've filled up numerous hard drives in a blink of an eye by saving to disk.

1.4 What is the purpose of a decode?

The purpose of a decode is to break down the packet fields and translate the field values into understandable form. For example, in Figure 1.1 the value 0x0800 in the Ethertype field was translated into 'IP.'

1.5 Why do you need a promiscuous mode card on your network analyzer?

You need a promiscuous mode card on your network analyzer to enable you to capture error packets and packets that are addressed to other devices on the network.

1.6 What are the primary elements of an analyzer?

Some of the primary elements of an analyzer are the ports, decode, capture filter, display filter, trace buffer, gauges and graphs and alarm system. Distributed analyzers also have agents or probes that gather the packets at a remote location and send them to the manager console.

1.7 What are the three ways of analyzing a switched network?

The three ways of analyzing a switched network are to hub out, use an analyzer agent on the switch or connect to a monitor port/SPAN ports.

1.8 What format are communications shown in when an analyzer does not have a decode for the protocol?

Typically communications are shown in hexadecimal format when an analyzer does not have a decode for the protocol seen.

1.9 What is the value of the basic broadcast address?

The value of the basic broadcast address is 0xFF-FF-FF-FF-FF-FF. Ok, some of you token ring folks may think that was a trick question because token ring has a special 'token ring broadcast' address of 0xC0-00-FF-FF-FF-FF (always have to be a little different, don't you)?

1.10 What is a cyclical trace buffer? How would you use it for troubleshooting?

A cyclical trace buffer is one that uses FIFO (first in, first out) to cycle packets through the buffer and run continuously. In troubleshooting, you can leave an analyzer running continuously in a closet

and stop the capturing process when a problem occurs.

Chapter Two Answers

2.1 What is the difference between the absolute and relative timestamps?

Absolute timestamps provide the date and time that the packets arrive in the trace buffer based on the local system clock. Relative timestamps provide the time the packets arrived in the buffer based on the first packet received or a specific marked packet.

Relative timestamps are particularly useful when you are timing specific processes, such a the time between user requests and server replies.

2.2 You have caught an unrecognized multicast packet in your trace buffer. Where should you go to lookup the registration for this address?

Multicast address assignments are listed online at www.iana.org. Select Protocol Numbers Assignment and then “M” for multicast address.

2.3 Your network uses full-duplex hubs and switches. How can you connect your analyzer to ensure that you capture all traffic that flows through the TX and RX channels?

In order to capture both RX and TX transmissions, you need a special ‘tap’ device. This device reroutes copies of all RX and TX communications down one combined RX channel to the analyzer.

2.4 You want to capture all the traffic to and from your key router as you download music using Napster (or a similar music-sharing program). How does this task affect the buffer size?

Wow! Downloading 2MB, 4MB, 6MB and greater files can really fill up a buffer quickly. If you are interested in catching some of this traffic without filling up your buffer, consider setting up your analyzer to slice the packets -- only capture a portion

of each packet. You don't need the entire data that follows the headers anyway.

2.5 Analysis Test

EtherPeek users: Download *ftp-file.pkt* from www.packet-level.com/tracefiles**.**

Sniffer users: Download *ftp-file.cap* from www.packet-level.com/tracefiles**.**

What format is the source address in (MAC-address, network address or name)?

The source addresses are listed as IP addresses.

What application is used in Packet 1?

Packet #1 is sent to port 21, the FTP command port.

What is the length of this packet?

This packet is 86 bytes long.

Chapter 3 contains some packet-by-packet analysis examples. Additional examples may be found online at www.packet-level.com.

Chapter Three Answers

3.1. A TCP/IP packet is addressed to the destination port number 53. What can you assume about this packet?

This packet is being sent to a DNS server. In DNS communications, the client uses a dynamic source port number and the destination port number 53.

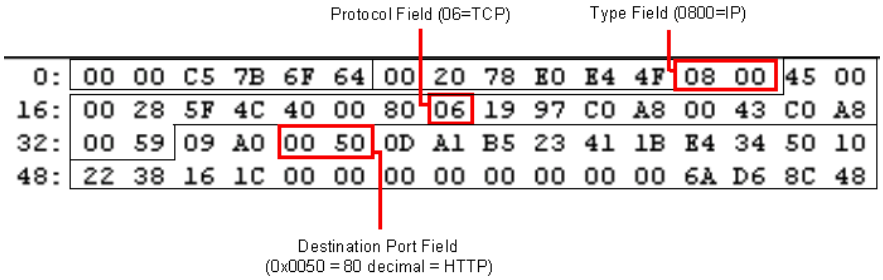
3.2. You have noticed a TCP communication that uses a Window field setting of 0. What does this indicate?

When the TCP window field is set to 0, the sending host is indicating that it has no available buffer space.

3.3. You are interested in capturing only ARP traffic on a network. What Ethernet type value should you filter on?

ARP traffic uses Ethernet type 0x0806.

3.4. OK... this one may take a couple of minutes... You are working with an analyzer that does not decode many packet types. Ugh. The following graphic provides an example of an undecoded packet. What can you figure out using the header structure information and sample headers shown in this packet?



The major fields are highlighted in the above figure.

The value 0800 in the Ethernet type field indicates that the IP header follows the Ethernet header.

The value 06 in the Protocol field indicates the next header is TCP.

The value 0050 (hexadecimal) in the destination port field translates to 80 decimal. Looking up the port number list at www.iana.org, we can see that this port is assigned to HTTP communications.

One way to check out the offsets is to open up a trace file and scroll through a packet, watching where the offsets reside in the hexadecimal portion. -- Laura

Chapter Four Answers

4.1 You are analyzing an Ethernet network that is experiencing a high number of collisions, but relatively low utilization peaks. What could be the problem?

If an Ethernet network experiences a high number of collisions but relatively low utilization peaks, you should know that the collisions are not a result of network congestion. Most likely, this network has a faulty device. Perhaps a network interface card is not deferring when the line is busy.

4.2 Where can you get the CSMA/CD specification?

You can get all the IEEE specifications online from www.ieee.org. I've also seen them at Computer Literacy (their online bookstore is at www.fatbrain.com - no kidding!).

4.3 What is the difference between an Ethernet collision fragment and a short packet?

A fragment is less than 64 bytes with a bad CRC (just a little piece of junk), whereas a short packet is less than 64 bytes with a good CRC (not junk, just a shrimp).

4.4 You are analyzing a Token Ring network that is experiencing a high number of line errors reported from the server's address. How can you fix this problem?

A high number of line errors reported from a single address (such as the server's address, in this example) indicates a possible faulty network interface card on the reporting device. Replace the server's card and run another trace.

4.5 You have captured a soft error report MAC frame that reports a token error. Based on this packet, what do you know about the reporting station?

The transmitting device is the Active Monitor because only the Active Monitor can send a soft error report MAC frame that reports a token error.

4.6 You see hundreds of SAP broadcasts on your network. Inside the IPX header, you notice that the source network address is 0x00-00-00-00 on these packets. What is most likely the purpose of these packets? How do you eliminate them from the network?

Since the source network address is 0x00-00-00-00 on these packets, we know the local system has not initialized the frame type with IPX. We can figure that these frames are most likely caused by auto frame detection. Interestingly, Novell's client is partial to using SAP for auto frame detection, whereas Microsoft's client is partial to RIP. Explicitly select a single frame type and remove any reference to auto frame detection in the LAN driver configuration, client configuration and IPX/SPX stack configuration to eliminate this unnecessary broadcast traffic.

4.7 What is the most wonderful protocol of the entire TCP/IP protocol suite (speaking from an analyst's perspective, of course)?

The most wonderful protocol of the entire TCP/IP protocol suite (speaking from an analyst's perspective, of course) is ICMP. DHCP runs a distant second.

4.8 Define the difference between UDP and TCP.

UDP is a connectionless protocol. TCP is a connection-oriented protocol. TCP headers are larger than UPD headers and offer some sequencing

and acknowledgment fields to provide end-to-end reliable transport.

4.9 Define the three packets seen in the TCP handshake.

The three packets seen in the TCP handshake are (1) SYN, (2) SYN ACK, and (3) ACK.

4.10 How can you perform a test to see if IP fragmentation is required to communicate between your client on a Token Ring network and a server on a remote Token Ring network?

You can perform a test between devices with the ping command. Use the ping command with the -f (do not fragment) parameter along with the -l (packet size to send) command. Set the packet size to the typical maximum packet size used on your Token Ring network (4096 for example).

APPENDIX B

The Basic Flow of Data

This information is MUST READ for anyone who works on networks. It's not a lot to go through and it is required reading - find a nice spot where no one will bother you for 30 minutes. Really concentrate and make notes to yourself so you remember the information.

Here's a hint on how to retain this information a bit better - I'm sure you have read zillions of books and articles and they are just a blur in your mind. Make extensive notes. The process of getting this information from your brain to your hand will leave a little groove (or scar) inside your arm and you'll remember it better from the painful process of note taking. Seriously folks - if you want to remember something, try to write it down as you learn it. Try to make a drawing - try to explain it to someone else. There's no better way to learn than to teach! -Laura

The Damned OSI Model Revisited

Ok, ok… you really can't avoid the OSI (Open Systems Interconnect) model. No matter how much you try. We won't focus on the entire model, however - we'll only focus on the layers that deal with the flow of data through an internetwork.

Figure B-1 shows the OSI model and the interconnecting devices that must abide by the rules of that layer.

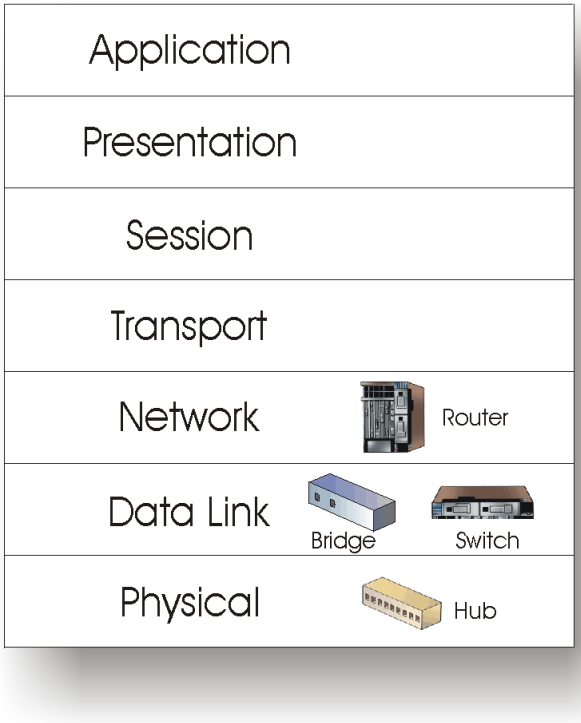


Figure B-1: The OSI model and the interconnecting devices.

In the next section, we'll examine each of the devices shown and check out how the communications flows through those devices.

Basic Hub/Repeater Communications

As you can see, hubs (and basic multistation access units and repeaters) are the most basic forwarding devices. They forward bits from one port to another. All they see are 1s and 0s. They really aren't very smart. A packet goes in one port of a hub and it is copied down all other ports of a hub.

This, of course, can be very ugly if one device is sending out a broadcast storm. It will affect all other devices connected to the same hub. So, you decide to control the traffic with another device - a bridge or a switch.

Basic Bridge and Switch Communications (Layer 2 Devices)

Bridges and switches forward based on the destination MAC address. They learn where devices are located when the devices initially communicate on a network. They put this address information in a table, as shown in Figure B-2.

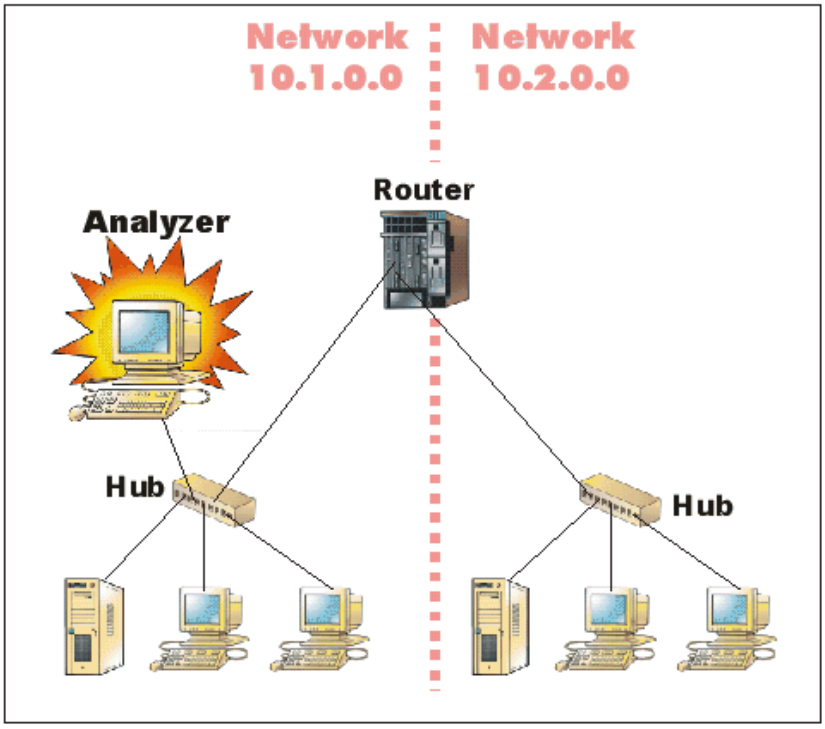


Figure B-2: Switches forward based on the device MAC address.

As you can see from Figure B-2, the contents of the packets do not change when they go through a switch (or a bridge). These devices do not change the network address information or the MAC address information.

Interestingly, broadcasts and multicasts are forwarded out all ports (because, after all, they are addressed to the group or entire set of devices).

These days, it would be very tough to buy a bridge -they have been replaced by switches. And many switches are beginning to 'cross-dress,' acting more like routers than switches. -Laura

Basic Router Communications (Layer 3 Devices)

Routers make forwarding decisions based on the network layer addressing contained in the packet. This information is contained in the IPX and IP headers, for example.

Local devices address packets to the router's MAC address in the MAC header. Upon receipt of the packets, the router must perform the following four steps:

1. *Check the Incoming Packet for Corruption; Remove the MAC header*

   Upon receipt of the packet, the router checks the packet for MAC-layer errors. The router strips off the MAC header and examines the network layer header to determine what to do with the packet.

2. *Examine the Age of the Packet*

   The router must first determine that the packet has not come too far to be forwarded on.

   IPX headers, for example, contain a hop count. By default, 15 hops is the maximum number of hops (or routers) that a packet can cross. If a packet comes in with a hop count of 15 in the IPX header, the router will discard the packet.

   IP packets contain a Time to Live (TTL) value in the IP header. Unlike the IPX hop count that increments as the packet is forwarded through each router, the IP TTL value decrements as the IP packet is forwarded through each router. If an IP packet comes in to a router with a TTL value of 1, the router will discard the packet. A router cannot decrement the value to 1 and forward it on.

3. *Determine the Route to the Destination*

   These routers maintain a routing table that lists available networks and the outgoing interface and distance to those networks. Once the router has determined which direction to forward the packet, it must build a new header.

4. *Build the New MAC Header and Forward the Packet*

   Finally, the router will build a new MAC header that addresses the packet from the router's MAC address to the final destination's MAC address or the MAC address of the next router in the path.

Figure B-3 shows the contents of a packet before and after it has gone through a router. It also shows the contents of the router's routing tables.

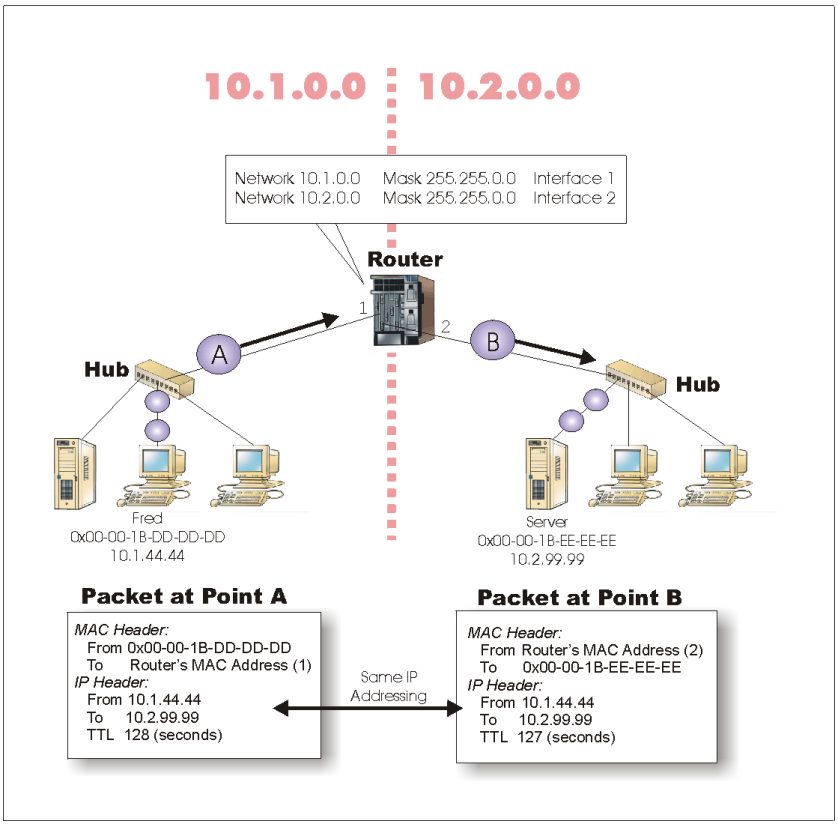


Figure B-3: Routers forward packets based on the network address.

Go ahead - capture the packets on each side a router on your network. You should be able to see the change in the hop count or TTL value and the new MAC header. This is key - if you ever want to analyze a communication, look at the network layer to determine the actual source and destination of the packet.

Exceptions to the Rules...

Nowadays, switches and routers are taking on a 'cross-dressing' role. Routers can switch data from one port to another without accessing their primary routing tables. Switches can route data when they do not have the MAC address in their tables.

The two devices are melting together to create what we now call, "Layer 3 switches" or "switching routers." These devices typically switch whenever possible (an entry exists in the MAC address table) and route whenever necessary (no MAC address exists in the table and the packet is addressed to another network).

APPENDIX C

Analysis References and Resources

This section lists a variety of resources including web sites, articles, and books.

Hardcopy Book Notes

If you purchased the hardcopy version of this podbook, you received "Laura's Lab Kit" on the CD.

Downloaded Book Notes

If you downloaded this podbook, you did not receive Laura's Lab Kit. You can, however, order the Lab Kit separately at www.pod-books.com. If you don't want to order Laura's Lab Kit, you can download the demo software from each of the vendors listed in Chapter 5.

Multiple Book Bundle

The 4-Book CD includes Laura's Lab Kit on the CD with the books.

Greypapers, Whitepapers and Articles

Greypapers are basically 'whitepapers with an attitude.' They address issues head-on with a bit of a smirk.

- Chappell, Laura. Auto-Frame Typing is Evil! Assign Your Frame Types Manually; 1998 [www.packet-level.com]
- Chappell, Laura. Why You Should Avoid the Ethernet_802.3 Frame Type; 1998. [www.packet-level.com]

The following articles written by Laura Chappell were published in NetWare Connection Magazine and can be downloaded/viewed at www.nwconnection.com.

- A NetWare Tool Time. (Sept. 1996, pp. 58-60)
- Ethernet Switches: Faster Than a Speeding Hub. (Aug.1996, p. 50-52)
- Houston NetWare Conference and Exhibits. (May 1996, pp. 61-62)
- IntranetWare Over TCP/IP. (June 1997, pp. 30-39)
- Inside DHCP. BrainShare '99 Conference Daily; March 25, 1999
- IPX-IP Gateways: Find Internet Solutions at a NetWare Conference and Exhibits. (July 1996, p. 60-64)
- Is your Network Connected? Finding Out Which Network Devices Are Responding. (March 1998, pp. 16-24)
- Is Your Network DOOMed (Jan./Feb. 1996, pp. 6-18)
- ManageWise 2.0: Novell's Management Platform Comes of Age with D. Hakes (Nov./Dec.1995, pp.51-56)
- Migrating to Pure IP: With NetWare 5. (September 1998, pp. 34-36)
- Mobile IPX: Unshackle Your Computer. (Aug. 1996, p. 24-32)
- Multimedia: It's Not Just for Video Games Anymore with Dan. Hakes (Sept./Oct.1994, pp.8-24)
- Multimedia: Making It Work on the Network (Nov./Dec. 1994, pp. 34-42)
- NetWare Link Services Protocol: Building a Link State Database. (Sept. 1997, pp. 38-45)
- NetWare Link Services Protocol: Interoperating With RIP and SAP. (Nov. 1997, pp. 36-39)
- NetWare Link Services Protocol: Updating the Link State Database. (Oct. 1997, pp. 34-39)

- Novell's NetWare Web Server: An Easy Way to Stake Your Claim in Cyberspace (June 1996, pp. 6-18)
- On the Road Again: Technical Education at NetWare Conferences. (June 1996, p. 51)
- On the Wire Again... (July/Aug. 1995, pp. 34-43)
- Onsite Network Analysis. (April 1999, pp. 28-31)
- Onsite Network Analysis. BrainShare '99 Conference Daily; March 21, 1999, pp. 3-9.
- Preserving WAN Bandwidth NetWare Connections. (Aug.1997, pp. 32-37)
- Pure IP Architecture. BrainShare Conference Daily, April 1998, pp. 3-7
- Service Location Protocol: Discovering Services in a Pure IP Environment. (July 1998, pp. 32-36)
- TCP/IP Troubleshooting Tools. (January 1999, pp. 20-28)
- Troubleshooting: Analyzing and Optimizing NetWare Communications. (Mar./Apr. 1994, pp. 12-16)
- Troubleshooting: Identifying and Eliminating Problems on Ethernet Networks. (Nov./Dec. 1993, pp. 32-34)
- Troubleshooting: Identifying and Eliminating Problems on Token Ring Networks. (Jan./Feb. 1994, pp. 44-48)

Technical Books

- Buchanan, Jr., R.W. The Art of Testing Network Systems. John Wiley & Sons, Inc.; 1996.
- Chappell, L., Advanced Cisco Router Configuration. Cisco Press and Macmillan Technical Publishing, Inc.; 1998.
- Chappell, L., Advanced Network Analysis Techniques. Podbooks.com; 2001.
- Chappell, L., Cisco Internetwork Troubleshooting. Cisco Press and Macmillan Technical Publishing, Inc.; Publication pending.
- Chappell, L., Hands-On Cisco: Basic TCP/IP LAN Configurations. Podbooks.com; 2000.
- Chappell, L., Introduction to Cisco Router Configuration. Cisco Press and Macmillan Technical Publishing, Inc.; November, 1998.
- Chappell, L., Novell's Guide to LAN/WAN Analysis: IPX/SPX. IDG Books Worldwide, Inc.; March 1998.
- Chappell, L., TCP/IP Analysis and Troubleshooting. Podbooks.com; 2000.
- Comer, D., Stevens, D. Internetworking with TCP/IP, Vol. II; Design, Implementation, and Internals, 4th Edition. Prentice Hall; 2000.
- Graham, B., TCP/IP Addressing. Academic Press; 1997.
- Haugdahl, S., Network Analysis and Troubleshooting, Addison-Wesley Pub Co.; 2000.
- Hein, M., Griffiths, D., Switching Technology In the Local Network: From LAN to Switched LAN to Virtual LAN. International Thomson Computer Press; 1997.
- Huitema, C. IPv6 The Internet Protocol, 2nd ed. Prentice Hall PTR; 1998.
- IEEE Technical Committee. Local and Metropolitan Area Networks. 802.1D; Media Access Control (MAC) Bridges. Institute of Electrical and Electronics Engineers, Inc., March 8, 1991.
- IEEE Technical Committee. Local and Metropolitan Area Networks. 802.1E; System Load Protocol. Institute of Electrical and Electronics Engineers, Inc., March 8, 1991.
- IEEE Technical Committee. Local and Metropolitan Area Networks. 802.3 Supplements; Layer Management (Section 5). Institute of Electrical and Electronics Engineers, Inc., March 8, 1991.

- IEEE Technical Committee. Local and Metropolitan Area Networks. 802.5 Supplements; Practice for Use of Unshielded Twisted Pair Cable (UTP) for Token Ring Data Transmission at 4 Mb/s. Institute of Electrical and Electronics Engineers, Inc., October 18, 1991.
- IEEE Technical Committee. Local and Metropolitan Area Networks. 802.5 Supplements; Recommended Practice for Dual Ring Operation with Wrapback Reconfiguration. Institute of Electrical and Electronics Engineers, Inc., September 18, 1991.
- IEEE Technical Committee. Local and Metropolitan Area Networks. 802.5; Token Ring Access Method. Institute of Electrical and Electronics Engineers, Inc., 1989.
- IEEE Technical Committee. Local and Metropolitan Area Networks. 1802.3 Conformance Test Methodology; Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications. Institute of Electrical and Electronics Engineers, Inc., September 20, 1991.
- Kwok, T. ATM The New Paradigm Form Internet, Intranet & Residential Broadband Services & Applications. Prentice Hall PTR, A Simon & Schuster Company; 1997.
- Perlman, Radia. Interconnections Bridges and Routers, 2nd Edition. Addison-Wesley Publishing Company, Inc., 1999.
- Sacket, George C., IBM's Token-Ring Networking Handbook. McGraw-Hill, Inc.; 1993.
- Seifert, Rich, The Switch Book: The Complete Guide to LAN Switching Technology. John Wiley & Sons; 2000.
- Sidhu, G., Andrews, R. Oppenheimer, A. Inside AppleTalk, 2nd ed. Apple Computer, Inc.; 1990.
- Spock, M.D., B., and Rothernberg, M.D., M.B. Dr. Spock's Baby and Child Care. Simon & Schuster; 1992.
- Stallings, Ph.D., W., Data And Computer Communications, 6th ed. Macmillan Publishing Company; 1999.
- Stevens, R. TCP/IP Illustrated, Vol. 1 The Protocols; Addison-Wesley Professional Computing Series; December 1994

Index

Numerics
10Base-T 26

A
abort delimiter 109
absolute time 57
absolute time column 59
Active Monitor 107
active stations 60
addressing problems 118
Advanced Network Analysis Techniques 9, 68
alarms 10, 25, 28
alerts 10
alignment errors 104
analyzer agent 24
analyzer agents 20
analyzer placement 16
application launch sequences 58
application testing 8
archives 68
ARI/FCI errors 109
ARP 54, 60, 73, 112
ASCII interpretation 69
attack 7
audio tours 4
authentication 5
authentication faults 11
auto framing problems 125

B
bad patterns 47
bandwidth 11, 18, 46
bandwidth requirements 8
baseline 60
Bay Networks/Nortel 20
beaconing 107
bit bucket 30
bits per second 11
bootup sequence 62
bootup traffic 11
bridged network 16, 17
Broadcast 55
broadcast 27
broadcast storms 29, 45
broadcast traffic 2, 3, 4
broadcasts 29, 61, 125
buffer size 38
burst errors 109
burst mode 61
busy network 3

C
cable faults 11
Callon, Ross 6
capture filter 3, 4, 25, 27, 38, 60
capture filters 25
card faults 11
Catalyst 2900XL 21

Chappell, Laura A. iii
Cinco Systems 42
Cisco 20
cold and hard floors 12
collisions 103
command pattern 47
communication lost 5
communication patterns 11
configuration faults 11
congested network 11
copyright@podbooks.com ii
CRC 57, 73, 82, 108
CRC errors 104
CSMA/CD (Carrier Sense Multiple Access/Collision Detection) 101
cyclical mode 39

D

data link errors 28
data offsets 68
decodes 10, 25, 26, 67
default gateway 7, 118
delta (interpacket) time 58
delta (or interpacket) time 57
Denial of Service 121
denial of service 5, 60
DHCP 60, 82
DHCP (Dynamic Host Configuration Protocol) 26
DHCP problems 120
discovery processes 7
display filter 3, 4, 30
display filters 25, 27
distributed analyzers 12, 13
Distributed Sniffer 13
DNS (Domain Name System) 112
documentating your network 49
downloaded book 131
duplicate requests 47
dynamic port number 69
dynamic port numbers 85

E

EGP 81
EIGRP 82
encapsulation 22
error checking 82
error packet 54
error packets 11
error rate 61
EtherLink III MLID 11
Ethernet header 10
Ethernet II frame 72
Ethernet packet structures 57
EtherPeek 15
EtherPeek Demo 132
excessive broadcasts 7, 45
excessive SAP/RIP broadcasts 124

F

failure replies 7, 45
FIFO (first in, first out) 30
file transfer pattern 47

filtering 2
flow of data 2
flying pigs 6
fragmentation 80
frame copied error 109
frame type 5
frame type mismatches 47
frequency error 110
FTP file transfer 9
FTP response code 68
full-duplex connections 24
full-duplex taps 24

G

gauges 27
general alarms 2
global statistics 2
graphical capabilities 42

H

hardware address 16, 54
hardware/software analyzer 15
header checksum 82
hex window 67, 69
hexadecimal 10, 26
host name 111
hot spots 44
HTTP 54, 91
HTTP Ccmmunications 94
HTTP servers 2
HTTP web browsing 9
hub 2
hubbed network 16
hubbing out 19

I

IANA 82, 114
ICMP 61, 81, 113, 114
  Destination Unreachable 115
  Parameter Problem 116
  Redirect 115
  redirection 7
  Time Exceeded 116
IDRP 82
IEEE 72
IEEE 802.3 74, 104
IEEE 802.5 76
IGMP 81
IGRP 56, 81
iNetTools Demo 132
info@podbooks.com ii
Institute of Electrical and Electronic Engineers (IEEE) 31
internal errors 108
Internet Assigned Numbers Authority (IANA) 31
Internet Engineering Task Force (IETF) 31
Internet Order of Old Farts 6
IP 73, 78
IP address 26
IP datagram 71
IP flags 80
IP fragmentation problems 119

IPCalc Demo for Palm 132
IPCalc for Windows 132
IPX/SPX 27, 61, 73, 74, 122

J
jabber 105

K
keep-alive sequences 45

L
Lab Exercise
  Capturing Packets 133, 135
  Checking out the Path to Podbooks.com 133, 141
  Checking Out Your Protocols and Packet Sizes 133
  Checking out your Protocols and Packet Sizes 136
  Converting an HP Advisor Trace File 133, 151
  Defining Subnets for 191.15.0.0 133
  Defining subnets for 191.15.0.0 158
  Explore Domain Information for laurachappell.com 133
  Explore Domain Information of laurachappell.com 155
  Opening Saved Trace Files 133, 138
  Scrubbing IP Addresses from an HTTP Trace File 133, 148
  Using WhoIs and Learning to Gather Information 133
  Using WhoIs and learning to gather information 144
LAN driver 71
LANalyzer for Windows 4
late collision 103
Laura's Lab Kit 131
layer 2 switches 18
layer 3 switches 18
laying on of hands 35, 38, 49
lchappell@netanalysis.org iii
line rrrors 108
local collision 103
local system clock 59
login sequence 57, 62
long frames 104
long-term traffic patterns 10
lost frame 109

M
MAC header 71
Macintosh 42
mailing list iii
marketing pukes 11
marking packets 52
maximum Ethernet frame size 74
Media Access Control 71
media access type 7
media fault 5
memory 30
minimum packet size 72, 101
Monitor Contention 107
MSS 89
MTU 81
multicast 11, 56
multicasts 7
my favorite protocol 114

N
Napster 46
NCP 61, 123
NDIS drivers 11
NDS 123
NeoTracePro Demo 132
NetBIOS 5, 55
NetBIOS broadcasts 45
NetDoppler Demo 132
NetScanTools Demo 132
NetSense Demo 132
network address 54
network analysis session 2
network congestion 28
network layer header 70
NetXRay 42
NLSP 56, 61, 123
Novell's Guide to IPX/SPX Analysis 73

O
optimization 7
OSPF 56, 82, 123

P
packet sequences 95
packet size 53
packet size distribution 28, 57
packets per second 10, 11, 29
PacketScrubber Demo 132
padding 72, 74
Parallel Tasking technology 11
path location 5
pattern analysis 47, 48
payload 68
ping 120
podbooks.com ii
port mirroring 20
port numbers 56
port spanning 20
Postel, Jon B. 115
post-filters 27
pre-filter 3, 25
pre-filters 27
probes 13
ProConvert Demo 132
promiscuous mode 11
promiscuous mode card 25, 26
promiscuous mode driver 25, 26, 36
proprietary traffic 45
Protocol Analysis Institute 31
protocol faults 11

R
receipt time 54
receive congestion 109
redirect traffic 35
redirecting traffic 37
registered port numbers 85
relative time 57
remote agents 12, 13

remote collision 103
remote process 58
repeater 76
reply 5
request 5
RFC 1122 89
RFC 1925 6
RFC 1945 92
RFC 2131 27
RFC 2616 91
RFC 768 84
RFC 791 78, 83
RFC 792 116
RFC 793 70, 87, 89
Ring Error Monitor 107
Ring Purge 107
RIP 60, 85, 123
RIP broadcasts 45
RIP2 56
route location 5
route location faults 11
routed network 12, 21
routing problems 118, 124
routing protocol overhead 7

S

Sam Spade 132
SAP 60, 122, 125
SAP broadcasts 45
screenshot captioning 49
sequence number 87
service discovery 5
service discovery faults 11
service discovery problems 118, 124
short frames 104
SLP 56
Sniffer Basic demo 132
Sniffer Basic/Pro 15
software analyzer 15
SPAN (switched port analysis) 20
speed of light 6
SPX 123
standalone analyzer 12, 21
standalone analyzers 12
statistics 3
summary page 49
summary screen 39, 52
summary window 50, 67
switched network 18
switched networks 12

T

tapping in 37
TCP 69, 78, 81
TCP communication 46
TCP connection problems 119
TCP header structure 87
TCP segment 69
TCP window 40
TCP/IP Analysis and Troubleshooting 40
TCP/IP problems 111

temporary port number 69
testing 8
time to live 81
token error 110
Token Ring 76
Token Ring broadcast 55
token ring network 26
Token Ring problems 106
Token Ring speeds 107
TokenPeek Demo 132
too many broadcasts 44
trace buffer 3, 25, 27, 30, 42
trace files for download 4
trace files for practice 127
traffic patterns 7, 39, 50
transport layer header 69
trend graphs 10
trends 42, 43
triggers 30
troubleshooting 5
turkey analyzer 4
Twelve Networking Truths 6
Type of Service 79

U

UDP 69, 78, 82
UDP checksum 86
UDP header structure 84
ugly application 8
ugly frame types 126
unanswered requests 44
undecoded traffic 45
unnecessary traffic 45
unrecognized packet structures 44
unrecognized sources 44
unusual patterns 10
URL 92
utilization 10, 29, 60

W

WAN link 22
War Room 14
WatchPoint Lite Demo 132
WebStats Demo 132
well-known port numbers 84
WildPackets products 132
www.iana.org 31
www.ieee.org 31
www.ietf.org 31
www.netanalysis.org iii
www.packet-level.com 31
www.podbooks.com ii

Second Edition

Introduction to Network Analysis

Laura Chappell

Chapter 1: What is 'Analysis, Anyway?

This chapter covers the various elements of a network analyzer and provides instructions on tapping into hubbed, switched, or routed networks. In this chapter, you'll find out how standalone analyzers differ from distributed analyzers as well as the differences between software and hardware-based solutions.

Chapter 2: How Do You Start?

This chapter walks you through the entire process of tapping in, obtaining a network overview (the 'Laying on of Hands'), classifying traffic and trends, identifying hot spots and documenting your findings. You'll learn how to read the summary window and detailed packet decodes. I'll introduce my favorite analyzers -- EtherPeek and Sniffer Pro. Finally, this chapter gives you a list of seven analysis tasks you can do today on your network.

Chapter 3: Packet Formats and Decodes

This chapter takes you into the packet structures and communication sequences often seen on a network. In this chapter, we'll look the headers that are built to get packets from the application through the protocol stack and onto the network. Specifically, we'll examine Ethernet and Token Ring headers. This chapter also provides field offset and details for IP, TCP and UDP headers. Finally, we'll follow a typical communication to see how the data flows.

Chapter 4: Identifying Typical Problems

This chapter focuses on the most common errors that occur on Ethernet, Token Ring, IPX/SPX and TCP/IP networks. After a brief overview of the communications processes used by each network type, you are introduced to the types of problems that you should watch for. Throughout this chapter, you are provided with extra links to areas that contain more depth of information… for those of you who just can't get enough.

Chapter 5: Analysis Tools and Exercises

This chapter gives you an idea of the other tools that I use to analyze communications. These tools include my beloved NetScanTools Pro, PacketScrubber to sanitize packets, ProConvert to convert trace files between various analyzer versions, EtherPeek for Windows and Sam Spade. Of course, a good quality blender is also require to make the perfect Margarita!.

podbooks.com™

Other podbook titles available at www.podbooks.com:
"Advanced Network Analysis and Troubleshooting," ISBN 1-893939-286 <electronic/hardcopy>
"TCP/IP Analysis and Troubleshooting," ISBN: 1-893939-014 <electronic/hardcopy>
"Hands-On Cisco: Basic TCP/IP LAN Configurations," ISBN: 1-893939-030 <electronic/hardcopy>